**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO**

DESENVOLVIMENTO DE MODELO MATEMÁTICO E SIMULAÇÃO DE UM ROBÔ MANIPULADOR DO TIPO SCARA

JOÃO VICTOR DE OLIVEIRA

RHENAN DIAS MORAIS

GUARULHOS – SP

2018

**JOÃO VICTOR DE OLIVEIRA**

**RHENAN DIAS MORAIS**

DESENVOLVIMENTO DE MODELO MATEMÁTICO E SIMULAÇÃO DE UM ROBÔ MANIPULADOR DO TIPO SCARA

Trabalho de conclusão de curso orientado pelo Professor Me. Rogério Daniel Dantas, a ser apresentado no Instituto Federal de Ciência e Tecnologia de São Paulo, Campus Guarulhos como requisito básico para a conclusão do curso de Tecnologia em Automação Industrial.

GUARULHOS – SP

2018

*Pai se este cálice de sofrimento não pode ser afastado de mim sem que eu beba, então que seja feita a sua vontade.*

**RESUMO**

O termo robô tem origem etimológica na palavra tcheca *robota,* que significa *trabalho forçado*, e no imaginário popular sua imagem se aproxima da forma humana, na qual existe um autômato capaz de realizar as mesmas tarefas e de forma similar aos humanos. Entretanto, o conceito de robô se atualiza constantemente, e desde o inicio do século XX, com a necessidade de melhoria na qualidade dos produtos e aumento da produtividade, a robótica industrial ganha corpo e encontra suas aplicações. Com o crescente avanço computacional em *hardware* e *software*, a robótica recebe suporte para grandes saltos no desenvolvimento tecnológico e integração com ambientes industriais, sendo possível obter velocidades de processamento superiores às das ultimas décadas e maior possibilidade de previsão do funcionamento de um mecanismo robótico. Este projeto, objetiva desenvolver um modelo matemático para aplicação em um robô manipulador do tipo SCARA, simulando o funcionamento do mecanismo com a utilização da técnica *Hardware in the Loop* e o *software* de simulação de robôs *V-Rep*, em que a lógica de controle será embarcada em um ambiente controlado e suas respostas monitoradas virtualmente, sendo possível visualizar o funcionamento do manipulador por meio de um modelo tridimensional. Essa técnica permite ajustes durante o desenvolvimento do mecanismo robótico sem o risco de danificar uma instalação em ambiente real, com a característica de manter as respostas do sistema as mais próximas possíveis da aplicação final.

***Palavras chave:*** Robótica industrial; Modelo matemático; Hardware in the Loop; Manipulador SCARA.

**ABSTRACT**

The term robot has an etymological origin in the Czech word *robota*, which means forced labor, and in the popular imagination your image is close to the human form, in which there is an automaton capable of performing the same tasks and in a similar way to humans. However, the concept of robot is constantly updated, and since the beginning of the twentieth century, with the need to improve product quality and increase productivity, industrial robotics gains body and finds its applications. With the increasing computational advancement in hardware and software, robotics receives support for large steps in technological development and integration with industrial environments, being possible to obtain processing speeds superior to the last decades and greater possibility of prediction of the operation of a robotic mechanism. This project aims to develop a mathematical model for application in a robot manipulator of the SCARA type, simulating the operation of the mechanism using the Hardware in the Loop technique and the V-Rep robot simulation software, in which the control logic will be embedded in a controlled environment and the responses monitored virtually, being possible to visualize the operation of the manipulator by means of a three-dimensional model. This technique allows adjustments during the development of the robotic mechanism without the risk of damaging an installation in real environment, with the characteristic of keeping the system responses as close as possible to the final application.

***Keywords:*** Industrial robotics; Mathematical model; Hardware in the loop; SCARA manipulator.

**LISTA DE ILUSTRAÇÕES**

**LISTA DE TABELAS**

**LISTA DE ABREVIATURAS**

**SUMÁRIO**

[**1.** **INTRODUÇÃO** 11](#_Toc526189072)

[1.1. MECANISMOS ROBÓTICOS 12](#_Toc526189073)

[1.2. ROBÔ MANIPULADOR SCARA 13](#_Toc526189074)

[1.3. HARDWARE IN THE LOOP 14](#_Toc526189075)

[1.4. OBJETIVOS 15](#_Toc526189076)

[**1.4.1.** **OBJETIVO GERAL** 15](#_Toc526189077)

[**1.4.2.** **OBJETIVOS ESPECÍFICOS** 15](#_Toc526189078)

[1.5. JUSTIFICATIVA 16](#_Toc526189079)

[1.6. METODOLOGIA 16](#_Toc526189080)

[**2.** **FUNDAMENTAÇÃO TEÓRICA** 17](#_Toc526189081)

[2.1. ORIGENS DA ROBÓTICA 17](#_Toc526189082)

[2.2. ROBÓTICA NA FICÇÃO CIENTÍFICA 17](#_Toc526189083)

[**2.2.1.** **AS LEIS DA ROBÓTICA DE ISAAC ASIMOV** 18](#_Toc526189084)

[**2.2.2.** **CINEMA** 19](#_Toc526189085)

[2.3. BREVE HISTÓRICO DA ROBÓTICA 21](#_Toc526189086)

[**2.3.1.** **LINHA DO TEMPO** 22](#_Toc526189087)

[2.4. CINEMÁTICA INVERSA E DIRETA (DENAVIT-HARTENBERG) 25](#_Toc526189088)

[2.5. CONCEITO DE HIL (HARDWARE IN THE LOOP) 29](#_Toc526189089)

[2.6. SOFTWARE DE SIMULAÇÃO (V-REP) 32](#_Toc526189090)

[2.6.1. MOTORES DE SIMULAÇÃO 33](#_Toc526189091)

[2.6.2. PROGRAMAÇÃO EM LUA 33](#_Toc526189092)

[2.6.3. V-REP API 34](#_Toc526189093)

[2.7. HARDWARE DE IMPLEMENTAÇÃO E PROGRAMAÇÃO (ARDUINO) 36](#_Toc526189094)

[2.8. CONCEITOS DE ROBÓTICA 38](#_Toc526189095)

[2.8.1. DEFINIÇÃO DE ROBÔ 38](#_Toc526189096)

[2.8.2. ELEMENTOS DE UM ROBÔ INDUSTRIAL 39](#_Toc526189097)

[2.8.3. ELEMENTOS ESTRUTURAIS 40](#_Toc526189098)

[2.8.4. ATUADORES 41](#_Toc526189099)

[2.8.5. JUNTAS 42](#_Toc526189100)

[2.9. CLASSIFICAÇÃO DE ROBÔS 45](#_Toc526189101)

[2.9.7. ROBÔ DE COORDENADAS CARTESIANAS 45](#_Toc526189102)

[2.9.8. ROBÔ DE COORDENADAS CILÍNDRICAS 46](#_Toc526189103)

[2.9.9. ROBÔ DE COORDENADAS POLARES (ESFÉRICAS) 46](#_Toc526189104)

[2.9.10. ROBÔ DE COORDENADAS DE REVOLUÇÃO (ARTICULADO) 47](#_Toc526189105)

[2.9.11. ROBÔ SCARA 48](#_Toc526189106)

[2.10. PROGRAMAÇÃO DE ROBÔS 49](#_Toc526189107)

[2.11. MÉTODOS DE PROGRAMAÇÃO 50](#_Toc526189108)

[2.12. PROGRAMAÇÃO ON-LINE 51](#_Toc526189109)

[**2.12.1.** **PROGRAMAÇÃO POR APRENDIZAGEM** 51](#_Toc526189110)

[**2.12.2.** **PROGRAMAÇÃO POR LINGUAGEM TEXTUAL** 51](#_Toc526189111)

[2.13. PROGRAMAÇÃO OFF-LINE 51](#_Toc526189112)

[**3. METODOLOGIA** 52](#_Toc526189113)

[3.1. ESTRUTURA DO PROJETO 52](#_Toc526189114)

[3.2. MODELAGEM MATEMÁTICA 56](#_Toc526189115)

[3.2.1. MODELAGEM MATRICIAL 56](#_Toc526189116)

[**3.2.2.** **MODELAGEM TRIGONOMÉTRICA** 59](#_Toc526189117)

[3.3. MODELADEM EM PYTHON 63](#_Toc526189118)

[**3.3.1** **BIBLIOTECAS E DEPENDÊNCIAS** 63](#_Toc526189119)

[3.2.1. CINEMÁTICA DIRETA EM PYTHON 64](#_Toc526189120)

[**3.3.2.** **CINEMÁTICA INVERSA EM PYTHON** 65](#_Toc526189121)

[**3.3.3.** **TESTANDO O MODELO PROGRAMADO** 66](#_Toc526189122)

[**3.2.4.** **VISUALIZAÇÃO DA ÁREA DE TRABALHO** 69](#_Toc526189123)

[**3.2.5.** **TESTE DE POSICIONAMENTO** 72](#_Toc526189124)

[3.3. MODELO TRIDIMENSIONAL 74](#_Toc526189125)

[3.4. MODELAGEM EM ARDUINO 76](#_Toc526189126)

1. **INTRODUÇÃO**

Automação e robótica são duas tecnologias que se desenvolveram de maneira diretamente relacionada. Segundo GROOVER (1988), em um contexto industrial podemos definir a automação como uma tecnologia que se ocupa do uso de sistemas mecânicos, eletrônicos e computadorizados na operação e controle da produção. Assim, a robótica é classificada como uma forma de automação industrial.

A ideia de robótica está presente nas mentes dos seres humanos desde muito cedo, a partir do momento em que o homem percebeu que era possível construir coisas. Artesões da idade média já criavam máquinas com o objetivo de imitar o movimento humano, um dos exemplos são as estátuas na torre do relógio de São Marcos em Veneza, que batem o sino na hora, e as estatuetas do século XV que contam uma história ao lado do Old Town Hall Tower, no Relógio Astronômico em Praga. (Figura1).



(a) (b)

Figura 1: Estátuas na torre do relógio de São Marcos (a) e estatuetas no Old Town Hall Tower (b) (NIKU, 2013).

Embora, no imaginário popular, o conceito de robô represente uma figura que se aproxima da forma humana, este conceito se atualiza constantemente, e as aplicações de robótica ganham cada vez mais força em ambientes industriais. É preciso então, definir o que classifica um mecanismo como robótico, ou pertencente a outro segmento, como por exemplo, um manipulador.

* 1. **MECANISMOS ROBÓTICOS**

Não é possível intitular qualquer mecanismo automatizado como robótico. Tomando por exemplo a comparação entre um robô manipulador qualquer e um guindaste, é possível perceber uma diferença fundamental entre eles: o guindaste é controlado por um humano, enquanto o robô é pré-programado para executar uma tarefa. Apesar de exercerem a mesma função e serem bastante semelhantes, essa diferença determina se um dispositivo é um simples manipulador ou um robô.

Em geral, um robô é projetado e destinado a ser controlado por um computador, ou dispositivo semelhante, e tem seus movimentos redigidos a partir de técnicas que lidam com conceitos como graus de liberdade, cinemática, dinâmica e planejamento de trajetórias, e uma de suas notáveis vantagens incluem o fato de executarem movimentos precisos e eficientes, desde que projetados e modelados adequadamente.

Para NIKU (2013), robôs sozinhos quase nunca são uteis, eles são usados em conjunto com outros dispositivos, periféricos e outras máquinas de fabricação para executar uma tarefa ou fazer uma operação. O modelo de controle é parte integrante desse conjunto de dispositivos, devendo ser escolhido adequadamente com a estrutura do mecanismo a ser utilizado.

Quanto aos elementos e partes pertencentes a um robô, pode-se definir um mecanismo robótico como um conjunto de dispositivos que são integrados para formar um todo. Suas partes envolvem atuadores, responsáveis por executar os movimentos; sensores, responsáveis por coletar informações sobre o estado do robô; controlador, responsável por processar e controlar os movimentos; e atuador final ou extremidade, parte atuante do robô que irá conter a ferramenta ou elemento de manipulação coerente com o objetivo final da aplicação.

Tratando-se de modelamento de mecanismos robóticos, existem dois tipos de abordagem: a cinemática inversa, e cinemática direta. Como define CRAIG (2012), cinemática é a ciência que trata do movimento sem considerar as forças que o causam, são estudadas posição, velocidade e aceleração. As equações de cinemática direta podem determinar onde estará o terminal do robô caso todas as configurações de juntas sejam conhecidas, e a cinemática inversa permite calcular qual deve ser cada configuração a fim de localizar o robô em uma posição desejada.

* 1. **ROBÔ MANIPULADOR SCARA**

Uma das principais características que define um robô são os graus de liberdade que este possui, em outras palavras, a quantidade de movimentos individuais que o mecanismo pode fazer e que são usadas para localizar uma posição no espaço. A movimentação dos graus de liberdade se faz possível através das juntas que estão presentes no robô. Juntas são os elementos que ligam as partes rígidas do mecanismo robótico, chamadas de elos, e podem ter diferentes configurações de movimento, o que resulta em diferentes quantidades de pontos no espaço acessíveis ao manipulador.

A configuração SCARA, acrônimo de *Selective Compliant Assembly Robot Arm* (braço robótico de montagem com complacência seletiva), apresenta duas (ou três) juntas rotacionais paralelas, permitindo movimento em um plano, e uma junta prismática perpendicular a este plano permitindo um movimento vertical para o atuador final. A figura 2 mostra o esquema simplificado de configuração de um robô SCARA:

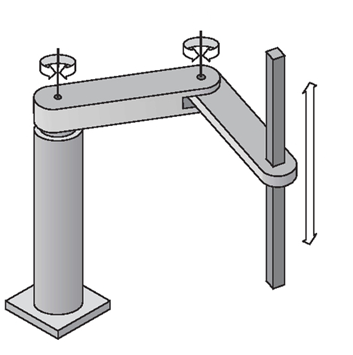


Figura 2: Esquema de configuração de um robô SCARA (CRAIG, 2012).

A aplicação de robôs SCARA é bastante comum em operações de montagem, uma vez que suas características são mais compatíveis com o plano *x-y*, mas com pouca flexibilidade ao longo do eixo *z.* Sua configuração e disposição de atuadores permite que o robô se movimente com muita rapidez, no entanto, com capacidade de carga útil reduzida.

Apesar de possuir área de trabalho menor do que alguns modelos de robôs articulados, apresenta diversas vantagens como velocidade, e relativa facilidade de programação do modelo cinemático, consequência de sua organização física de juntas.

* 1. **HARDWARE IN THE LOOP**

*Hardware in the Loop* (HIL), é um conceito que utiliza um sistema embarcado para simular o comportamento de um controlador em um ambiente real. É possível utilizar esta técnica para prever o comportamento de partes do sistema que apresentem algum desafio no desenvolvimento, buscando obter um comportamento o mais próximo possível da aplicação final, sem correr o risco de danificar partes atuantes do sistema durante testes de programação e desenvolvimento.

Um sistema com HIL é composto basicamente por uma interface para operação, um processador que executa as funções em tempo real, e um conjunto de entradas e saídas. O processamento de tempo real é normalmente necessário para a simulação exata das partes do sistema que não estão fisicamente presentes no teste. O esquema de um sistema HIL simplificado por ser observado na figura 3:

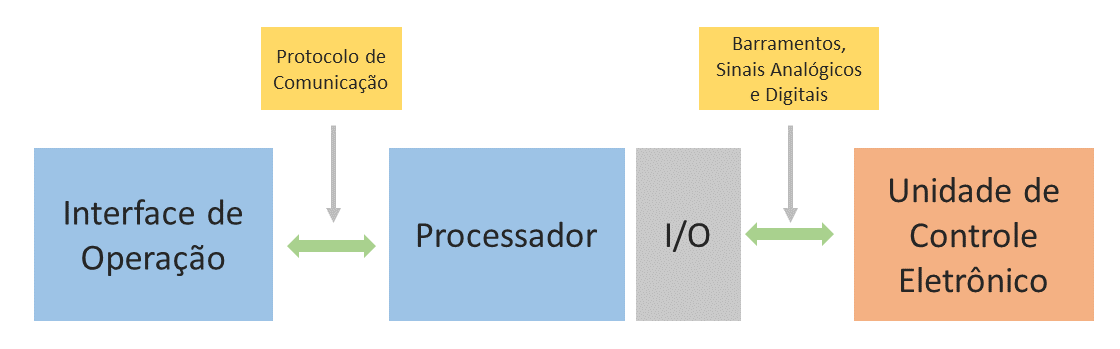


Figura 3: Esquema simplificado de um sistema HIL (National Instruments, 2016).

A interface do operador é utilizada para comunicação com o *hardware*, fornecendo visualização e monitoramento do sistema, além de possibilitar a execução de tarefas como gerenciamento da configuração, automação de testes, análise e gerenciamento de relatórios, entre outras. Os sinais de entrada e saída podem ser quaisquer sinais analógicos, digitais ou de barramentos, que interagem com a unidade em teste, e são utilizados para produzir sinais de estímulo, adquirir dados de registro e análise e fornecer as interações com sensores e atuadores entre a unidade de controle eletrônico (ECU) e o ambiente virtual do operador.

* 1. **OBJETIVOS**

Os objetivos traçados com o desenvolvimento deste projeto apresentam-se da seguinte forma:

* + 1. **OBJETIVO GERAL**

Desenvolver um modelo matemático para um robô manipulador do tipo *SCARA*, utilizando a técnica de *Hardware in the Loop* e softwares de simulação para monitorar o comportamento e as respostas obtidas com o modelo desenvolvido.

* + 1. **OBJETIVOS ESPECÍFICOS**
* Definição das equações necessárias para criação de um modelo matemático que descreva de forma satisfatória o movimento do manipulador;
* Modelagem tridimensional de um manipulador SCARA utilizando de softwares do tipo CAD (*Computer Assisted Drawing*);
* Integração do modelo tridimensional com o software de simulação de robótica V-Rep (*Virtual Robot Experimentation Platform*);
* Desenvolver programação embarcada do modelo matemático utilizando técnica *hardware in the loop;*
* Integrar programação embarcada do modelo com a simulação do mecanismo robótico.
  1. **JUSTIFICATIVA**

Um mecanismo robótico com estrutura física bem projetada e modelo matemático bem desenvolvido oferece possibilidade de controle preciso e eficiente. O processo de desenvolvimento de um robô, no entanto, não é algo simples e por vezes pode acarretar na danificação da estrutura robótica, ou do produto manipulado. Para evitar tais riscos, são utilizadas técnicas de simulação em ambientes controlados, sendo possível prever o comportamento das atividades em um ambiente real.

Deste modo, o projeto justifica-se em aplicar e disseminar o uso da técnica de simulação embarcada para um modelo de robô manipulador SCARA, utilizando o software V-Rep e a plataforma de desenvolvimento Arduino, ambas de fácil acesso e com ampla documentação disponível para o usuário, desenvolvendo durante o processo, ferramentas que podem ser adaptadas e facilmente utilizadas em outros tipos de robôs manipuladores, ou móveis.

* 1. **METODOLOGIA**

O método de desenvolvimento adotado tem caráter dedutivo, partindo das premissas comuns de controle de mecanismos robóticos, aplicando os conceitos especificamente em um robô do tipo SCARA. O estudo pretende-se vir a ser desenvolvido da seguinte forma:

* Pesquisa bibliográfica: levantamento do histórico da robótica, definições, classificações e aplicação no ambiente industrial;
* Desenvolvimento: desenvolvimento do modelo matemático e integração da plataforma embarcada com software de simulação;
* Análise dos resultados: verificação do controle de trajetória e posição obtido com o modelo matemático integrado ao software de monitoramento.

1. **FUNDAMENTAÇÃO TEÓRICA**
   1. **ORIGENS DA ROBÓTICA**

O âmbito da robótica tem origem na ficção científica, com primeiros relatos por volta de 1920. Um dos trabalhos mais relevantes para a discussão das origens da robótica é a novela escrita por Mary Shelley, em 1817, intitulada *Frankenstein*, que relata os esforços do Dr. Frankenstein em criar um monstro humanoide, fato que aterroriza a comunidade local. A imagem fixada na mente de milhões de pessoas durante o passar do tempo foi a do monstro criado pelo doutor, afastado das boas intenções de seu criador humano, que foi transportada para os robôs durante décadas.

O termo em inglês *robot* deriva da palavra tcheca *robota*, cunhada em uma peça de autoria de Karel Capek, intitulada *Os Robôs Universais de Rossum,* e significa servidão, ou trabalho forçado. Na história, Rossum, um cientista brilhante, começa a fabricar robôs que possuem a função de servir a humanidade obedientemente e fazer todo o trabalho físico necessário. O plano toma um rumo não desejado quando os seres quase perfeitos desenvolvidos pelo cientista começam a não gostar dos seus papeis de serventes e rebelam-se contra seus senhores, destruindo toda a vida humana.

Apesar das limitações dos mecanismos robóticos conhecidos atualmente, o conceito popular de robô é de que ele age como o ser humano, imagem que foi popularizada pela ficção científica, que também criou a ideia de superioridade dos robôs sobre seus criadores humanos.

* 1. **ROBÓTICA NA FICÇÃO CIENTÍFICA**

Não é algo impossível e sim a ser questionado, pois não temos algo concreto, tudo pode ser criado, porém temos que imaginar e especular robôs que tenham finalidades usuais, para não cairmos na ideia do R2D2 e do C3PO, que são os robôs mais famosos de todos os tempos, criações incríveis que o mundo já viu, na questão de conceito e usabilidade, executando funções humanas, reagindo a situações, até mesmo imitando sensações e sentimentos, para parecerem cada vez mais reais.

Os robôs tomaram um espaço gigante na vida das pessoas comuns, coisas que antes só eram vistas em laboratórios hoje são vistas nas casas, departamentos de segurança, sistemas de alarme, auxiliando pessoas com problemas motores, no tratamento de pessoas com depressão (robôs humanoides). Assim vemos cada vez mais um futuro que terá os robôs nos auxiliando em diversas tarefas, e quem sabe exercendo trabalhos manuais, domésticos e se multiplicando.

* + 1. **AS LEIS DA ROBÓTICA DE ISAAC ASIMOV**

Isaac Asimov, entre os escritores de ficção científica, contribuiu com inúmeras histórias sobre robôs, e possui o crédito de ter desenvolvido o termo *robótica* em uma de suas obras (Figura X). Em seu trabalho, a figura de robô é a de uma máquina bem projetada, livre de falhas e que atua de acordo com três princípios, chamados de *Três leis da Robótica:*

1. *Um robô não pode ferir um ser humano ou, por inação, permitir que um humano seja ferido.*
2. *Um robô deve obedecer às ordens dadas por humanos, exceto quando isto conflitar com a Primeira Lei.*
3. *Um robô deve proteger sua própria existência, a menos que isto conflite com a Primeira ou segunda Lei.*

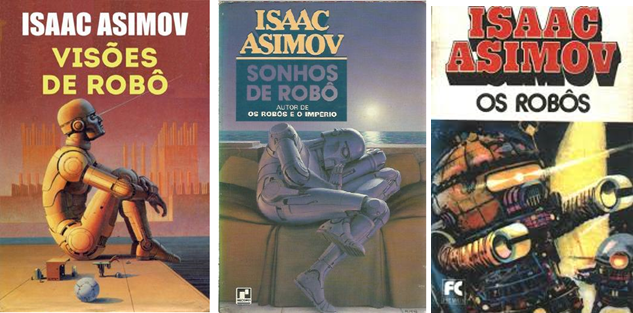


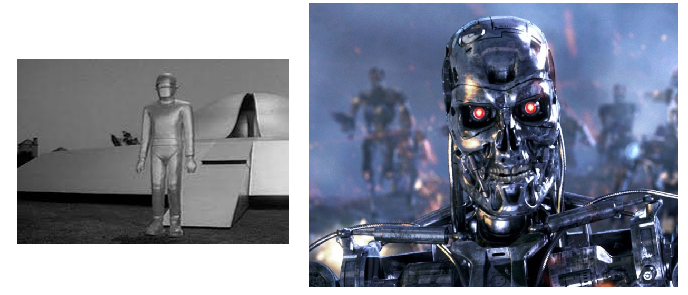
Figura X

* + 1. **CINEMA**

O conhecimento da robótica deve sua disseminação em grande parte a filmes e programas de televisão. Surge uma linha que se torna praticamente padrão para filmes de ficção: máquinas que foram feitas para ajudar as pessoas se rebelam e passam a lutar contra os seus criadores. Esse modelo de ficção científica é chamado por Asimov de “Complexo de Frankenstein”.

O filme *O Dia em que a Terra Parrou*, de 1951, é considerada uma das primeiras obras cinematográficas com a presença de robôs. Mostrava uma missão vinda de um planeta distante, enviada a Terra em um disco voador tripulado por um robô onisciente, onipotente e indestrutível, nomeado *Gort* (figura X). O robô era um pacifista universal, e, quando um planeta “saía da linha”, a punição era imediata e final, a obra demonstrou o terrível poder destrutível das futuras armas.

Já em *2001: Uma Odisseia no Espaço*, de 1969, a figura principal não era um robô mecânico, mas um computador inteligente, falante e com personalidade, chamado *HAL.* A tarefa do computador era controlar e monitorar os sistemas de uma espaçonave com destino ao planeta Júpiter. Durante o percurso, um dos circuitos de HAL falha, assim sua personalidade se deteriora e o computador começa a aniquilar os membros da tripulação.



1. (b)

Figura X: Robô *Gort* (a), e o robô *de O Exterminador do Futuro* (b).

Entre outros marcos da ficção que trabalham com a revolução das máquinas, pode ser citado *O Exterminador do Futuro*, de 1984, em que um robô revestido com tecido vivo volta no tempo para exterminar a maior ameaça aos próprios robôs do futuro, e *Matrix*, de 1999, em que as máquinas já tomaram o poder e suplantaram o mundo por um software de realidade virtual que mantem os seres humanos acoplados ao sistema.

Além da imagem de ser destrutivo e perigoso, robôs também foram apresentados ao publico de forma amigável. Na série *Guerra nas Estrelas*, iniciada em 1977, são apresentadas máquinas amistosas inofensivas. Os robôs R2D2 e C3PO (**figura X**), eram capazes de andar livremente e se comunicar com seus senhores humanos.

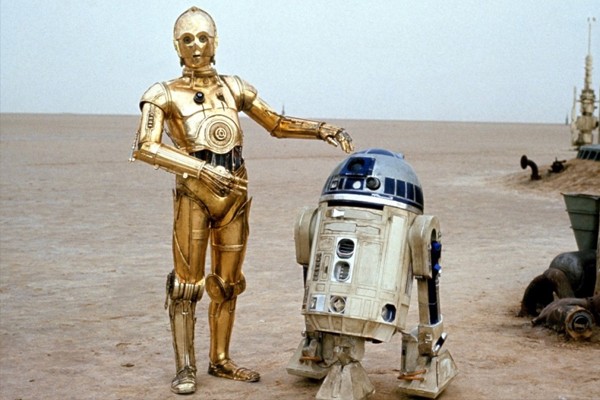


Figura X

Outro exemplo de tratamento da figura de robôs na ficção é aquela na qual a criação humana tenta se igualar a imagem de seu criador, adquirindo cada vez mais características humanas. Alguns exemplos são *O Homem Bicentenário*, de 1999, em que um robô que perdura por gerações, sendo moldado pelo seu entorno através do tempo, e *IA – Inteligência Artificial*, em que um robô semelhante a uma criança, capaz de desenvolver sentimentos, almeja se tornar um humano de verdade.

Observando assim o histórico da robótica na ficção e cinematografia, é possível traçar algumas conclusões de como a arte enxerga os robôs, androides, ciborgues ou qualquer outra criatura virtual, sendo possível identificar três vertentes:

* Robôs são criados de forma a facilitar as tarefas do dia a dia, sendo um amigo ou companheiro de seu senhor humano;
* Robôs são criados, ou se desenvolvem a tal ponto de objetivar a dominação da raça humana;
* Robôs são criados a imagem de seres humanos, buscando sempre se igualar ao seu criador, tornando-se o mais humano possível.
  1. **BREVE HISTÓRICO DA ROBÓTICA**

A criação ficcional dos robôs, seu deu com a fantasia de desenvolver uma máquina, análoga ao ser humano, capaz de pensar e agir de maneira inteligente, ainda que artificial, denotando uma vontade suprimida de ter uma espécie de “escravo metálico” que satisfaça todas suas vontades.

Apesar dos relatos antigos de criação de mecanismos robóticos, a ideia de construir robôs como são conhecidos hoje começou a tomar força no início do século XX, com a necessidade de aumentar a produtividade no meio industrial e gerar melhoria na qualidade dos produtos. Nessa época a robótica industrial encontra suas primeiras aplicações.

Como apresenta ROSÁRIO (2005), devido aos inúmeros recursos que os sistemas micro controlados nos oferecem, a robótica atravessa uma época de contínuo crescimento, que permitirá, em curto espaço de tempo, o desenvolvimento de robôs cada vez mais inteligentes, deste modo, talvez a ficção do homem antigo se torne realidade do homem atual.

* + 1. **LINHA DO TEMPO**

Há algumas décadas atrás, os robôs faziam parte apenas do imaginário humano e da ficção científica, e no início dos anos 60, começaram a ser utilizados para substituir o homem em tarefas que ele não podia realizar, as quais envolviam condições desagradáveis como alto nível de calor, ruídos, gases tóxicos, radioatividade, esforço físico extremo, entre outras.

Nos últimos anos, a evolução dos robôs tem sido garantida através de duas tendências: o aumento do nível salarial dos empregados e o notável avanço tecnológico, que provoca uma redução no custo de desenvolvimento de robôs e uma melhoria significativa em seu desempenho. A tabela X apresenta uma linha do tempo com alguns fatos históricos considerados marcantes para o desenvolvimento da robótica.

Tabela X: Alguns fatos históricos no desenvolvimento da robótica.

|  |  |
| --- | --- |
| Período | Fato histórico |
| Século IV A.C (Grécia) | Aristóteles relata os primeiros princípios similares aos conhecidos atualmente como robótica, utilizando instrumentos com o intuito de auxiliar determinados trabalhos. |
| Século XVIII | Com a Revolução Industrial, surgem novos mecanismos e instrumentos que tornam possível a evolução do maquinário e realização de uma série de ações sequenciadas. |
| Século XIX | Introdução do motor elétrico na industrial, máquinas começam a substituir tarefas manuais. |
| 1914 – 1918 | Primeira Guerra Mundial, o poder da máquina mostra sua forma negativa e destrutiva. |
| 1921 | A palavra *robô* é utilizada pela primeira vez na ficção. O texto relata a criação de robôs para substituir o homem nos trabalhos pesados. O robô começa a ser visto como uma máquina humana, com inteligência e personalidade, |
| 1940 | Isaac Asimov desenvolve e publica as *Três Leis da Robótica.* |
| 1946 | O inventor norte-americano G. C. Devol desenvolve um dispositivo controlador capaz de registrar sinais elétricos magneticamente e reproduzi-los para operar uma máquina mecânica. |
| 1951 | Operadores por controle remoto são utilizados pela primeira vez para manipulação de materiais radioativos, demonstrando a possibilidade de mecanismos robóticos em substituir o homem em tarefas perigosas. |
| 1959 | Devol e Joseph F. Engerlberger desenvolvem primeiro robô industrial pela *Unimation Inc*., capaz de executar automaticamente uma variedade de tarefas, além da possibilidade de ser reprogramado e remodelado para outras tarefas. |
| 1961 | O robô *Unimate*, de Devol, é instalado na *Ford Motor Company* para atender a uma máquina de fundição sob pressão. |
| 1969 | O homem pisa no solo lunar pela primeira vez. Já se utilizavam manipuladores robóticos para coleta de amostras e pequenas tarefas com comando por controle remoto. |
| 1974 | A *Cincinnati Milacron* apresenta o primeiro robô industrial controlado por computador, denominado *The Tomorrow Tool*, ou *T3*, era capaz de mover objetos em uma linha de montagem. O estudo sobre utilização de sensores ganha força. |
| 1974 | A *Kawasaki* instala robôs *Unimation* em operações de soldagem de arco para chassis de motocicleta. |
| 1975 | A Olivetti começa a utilizar o *Robô Sigma* em operações de montagem, uma das primeiras aplicações de montagem efetivas da robótica. |
| 1975 | Robô para montagem mecânica de uma máquina de escrever é desenvolvido na *IBM* por Will e *Grossman*. |
| 1978 | Robô *PUMA* (*Programmable Universal Machine for Assembly* - Máquina Universal Programável para Montagem) é desenvolvido pela Unimation, utilizando como base projetos de estudos do *General Motors.* |
| 1979 | Desenvolvimento do robô do tipo SCARA na *Universidade Yamanashi*, no Japão. Vários robôs SCARA são introduzidos comercialmente por volta de 1981. |
| 1980 | A *General Motors*, em Detroit, nos Estados Unidos, introduz um robô industrial com capacidade de reconhecer diferentes componentes em uma linha de transporte e escolher aqueles que necessita. |
| 1986 | Produtos educacionais baseados em LEGO são colocados no mercado. Honda lança um projeto para construir um robô humanoide que possa andar. |
| 1997 | O primeiro torneio *Robocup* é realizado no Japão, o objetivo é ter uma equipe totalmente automatizada de robôs para vencer o mundo a melhor equipe de futebol no ano de 2050. |
| 1999 | É lançada a primeira versão do cão robótico Aibo, da Sony, o cão robótico com a capacidade de aprender e se comunicar com seu dono. |
| 2000 | É lançado pela *Honda*, o *ASIMO*, primeiro da nova linha de robôs humanoides da empresa. |
| 2002 | É introduzido no mercado o robô *Roomba*, um aspirador robótico que rapidamente ganha sucesso e demonstra força da aplicação doméstica da robótica. |
| 2011 | Diversas empresas investem em estudos para a fabricação de exoesqueletos que podem ser integrados com um corpo humano a fim de ajudar pessoas com paralisia. |
| 2012 | Universidade de Pittsburgh desenvolve braço mecânico que pode ser controlado pelo pensamento |

* 1. **CINEMÁTICA INVERSA E DIRETA (DENAVIT-HARTENBERG)**

A fim de desenvolver um esquema para controlar o movimento de um manipulador, é necessário desenvolver técnicas para demonstrar a posição do braço em pontos no tempo. Definimos os robôs manipuladores por juntas e elos. Cada junta determina quantos graus de liberdade este robô há de ter, temos juntas com movimentos Lineares (junta tipo L) ou movimento Rotacional (juntas tipo R, T e V), entre os elos adjacentes. Os elos por sua vez são as estruturas rígidas que são ligados pelas juntas.

As juntas e os elos recebem letras e números para suas identificações, sendo as juntas rotuladas de , onde começa com 1 sendo a base do manipulador, e os elos , esse também começado em 1, sendo este o elo que estiver mais próximo da base. A notação , também é utilizada para indicar o comprimento de cada elo gerando assim derivações das equações.

A cinemática de um robô RR, mais complexa de analisar que a representação de um robô LL, começando através de um caso bidimensional, a posição da extremidade do braço pode ser representada de inúmeros modos. O modo mais usual é utilizar os ângulos das duas juntas e . Isto é conhecido como representação no espaço da “junta” e podemos defini-la como:

O outro modo de definir posição do braço é no espaço cartesiano. Envolver um sistema de coordenadas cartesianas que é extremo ao robô. A origem é determinada pela base do robô. A posição do fim do braço seria definida no espaço bidimensional como

Utilizado usualmente para comunicar o robô a outras máquinas que não tem uma compreensão detalhada da cinemática do robô, de modo que uma representação “neutra”, tal como o espaço cartesiano, tem de ser usada.

Para usarmos ambas as representações, devemos poder transformar de uma para outra. Ir do espaço de junta para o espaço cartesiano é chamado transformação direta e o mesmo acontece para o inverso, ir do plano cartesiano para espaço de junta, isso se chama “transformação inversa”.

A cinemática é o método utilizado para descrever matematicamente a movimentação de diversos tipos de mecanismos robóticos, com este método é possível controlar diferentes tipos de robôs através dos cálculos cinemáticos utilizando cinemática inversa ou direta.

No caso de robôs manipuladores, como o SCARA, na cinemática direta são conhecidas as posições das articulações e obtêm-se a posição da extremidade do manipulador e velocidades de movimento, já na cinemática inversa acontece o contrário da direta, dadas a posição da extremidade e velocidades, obtém-se as posições e velocidades correspondentes das articulações.

O método utiliza o conceito básico de matrizes de referência para encontrar o ponto desejado, aplicando cálculos com matrizes de referência (padrão), a fim de encontrar transformações de coordenadas, no caso do projeto desenvolvido, as coordenadas são descritas em um plano cartesiano, utilizando o conceito de *Denavit e Hartenberg*, que determina o movimento dos elos de movimento, que são as (hastes a serem controladas).

A cinemática como citado, utiliza matrizes para efetuação de cálculos, assim determinando a movimentação, posição, área de trabalho, utiliza também as próprias dimensões do robô, como, comprimento das hastes, altura do robô, ângulos de trabalho. Seguindo o conceito físico do nosso projeto que é um robô de duas articulações de revolução (2R), tem-se uma modelo base conforme observado na figura X:

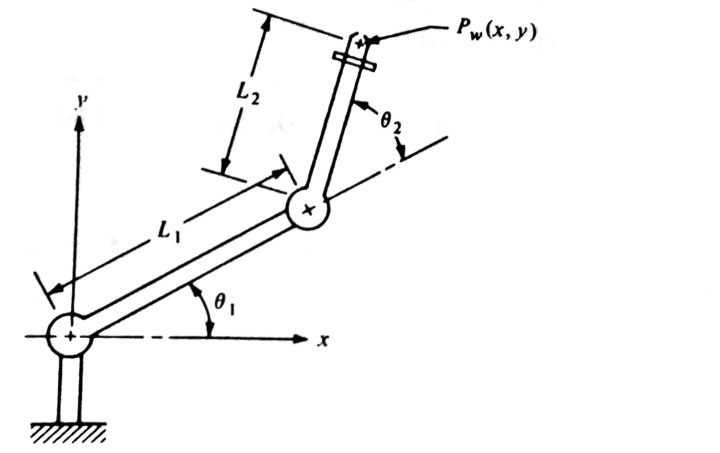


Figura X

A tabela X a seguir, apresenta a relação de elos e ângulos de movimento do modelo para um robô SCARA, seguindo os parâmetros de Denavit-Hartenberg.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Ligamento* |  |  |  |  |
| 1 | a1 | 0 | 0 |  |
| 2 | a2 | 0 | 0 |  |

O eixo Z do robô SCARA, não é uma das variáveis de cálculo, pois tem determinação independentemente do restante do robô, sendo possível implementar ferramentas, tanto pneumáticas, magnéticas, garras, de vácuo, entre outras.

As matrizes de transformação homogênea que determinam a posição final do robô são:

Sabendo que os dois primeiros elementos da quarta coluna são respectivamente *x* e *y* finais, assim já não mais necessitando a equação individual por hastes, com isso diminuindo o tempo de movimentação do braço incluindo apenas as equações finais na programação, observadas na expressão X a seguir:

* 1. **CONCEITO DE HIL (HARDWARE IN THE LOOP)**

*Hardware in The Loop (HIL)* é um conceito antigo criado em 1950 para simulação de uma situação real. O HIL foi desenvolvido para substituir ou servir de apoio para análises de um determinado sistema que é muito complexo e robusto para servir de teste, por temos caro, e eficiente, com complexos algoritmos, foi aplicado em sistemas de engenharia aeroespacial, recentemente essa ferramenta tem sido utilizada em diferentes ciclos industriais como sistemas automobilísticos, indústria, robótica, utilizando o HIL em laboratório é possível analisar o designer antes do final, simular e avaliar defeitos, praticar do antes do projeto final ser utilizado, garantindo a eficácia, qualidade e segurança do projeto que muitas vezes pode ser extremamente caro.

Um HIL ideal, consegue substituir um hardware e software exatamente implementados. Pois muitas vezes o diminui o custo drasticamente, pois algumas situações demandam precisão e preço baixo por isso o HIL é implementado.

O HIL implementado na robótica, os ambientes são possíveis de simulação, operação, podem ser controlados via internet, e os resultados podem ser trabalhados para ser colocada no HIL, a simulação pode ser controlada via software, pelo telefone celular, tudo aquilo que permite uma comunicação, robôs móveis e robôs manipuladores, respectivamente.

No HIL é permitido simular, explorar o modelo animado do robô, verificar, validar tudo isso por software e sistematizar a estrutura do robô. Controlando atuadores, carregando simulações e analisando o comportamento. Tudo isso é possível com HIL implementado.

Para sofisticados sistemas de controle utiliza-se o HIL, para assim obter um controle lógico e determinísticos, entretanto é possível controlar por exemplo o controle da temperatura de um quarto, até mesmo simular processos e situações de para uma aeronave.

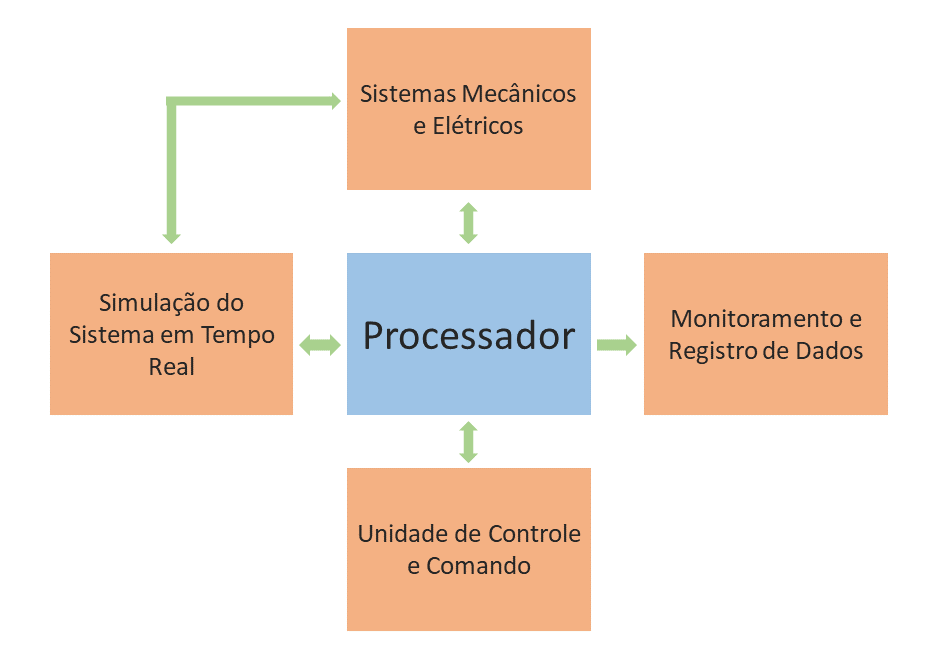
O HIL necessita de um nível alto de ordens mais precisão do que sistemas individuais de hardware ou software, o HIL é um conceito de *real-time,* simulação para testes embarcados nos controles de sistemas diversos, presentes como software e hardware de um computador, atuadores e sensores enquanto o sistema executa a simulação. De acordo a Tabela 1 o HIL, está presente como um dos mais convencionais sistemas de simulação.

No HIL ideal, todo sistema de hardware aceita comunicação com o HIL, exceto aqueles sistemas que possam ser substituídos por esse modelo. A única especificação desse modelo é que seja simulado com uma alta frequência o HIL oferece essa limitação, para o controle do *Loop,* isso porque o HIL necessita de um equipamento auxiliar para partição separada, indicado pela tabela de simulação

Tabela das possibilidades de simulação:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Linha | Modo | Sensor | Sistema | Atuadores | Computadores |
| 1 | Totalmente Simulado ou SIL | Simulado | Simulado | Simulado | Simulado |
| 2 | Parcialmente Simulado ou PIL | Simulado | Simulado | Simulado | Físico |
| 3 | Simulado com Atuadores | Simulado | Simulado | Físico | Físico |
| 4 | Ideal Hardware in the Loop | Físico | Simulado | Físico | Físico |
| 5 | Teste Prático | Físico | Físico | Físico | Físico |

Percebe que o projeto ideal parte de uma identificação de estrutura, exemplo é necessário a que saiba o que se deseja por desenvolvimento, onde terá atuação e como isso deve interferir no esquema ou projeto final, contudo temos uma estrutura padrão que auxilia a estruturação do projeto.



* 1. **SOFTWARE DE SIMULAÇÃO (V-REP)**

O V-REP, acrônimo para *Virtual Robot Experimentation Platform*, desenvolvido e publicado pela *Coppelia Robotics*, funciona como um laboratório virtual, para implementação e simulação de sistemas robóticos, onde é possível realizar a simulação de testes de movimentos, teste de força, resistência, etc., permitindo assim a visualização do comportamento de um mecanismo robótico, podendo modificar, programar rotinas, inserir defeitos e simular acontecimentos adversos à área de trabalho.

A robótica é um conceito multidisciplinar, abrangendo várias áreas do conhecimento, mecânico, elétrico, eletrônico, da informática e matemática. Pedagogicamente passando para os usuários, desafios e conhecimentos diversos, assim, o software atua integrando áreas de mecânica, programação e matemática.

A figura X apresenta uma imagem do ambiente de simulação do software com alguns robôs carregados e adicionados a uma cena.

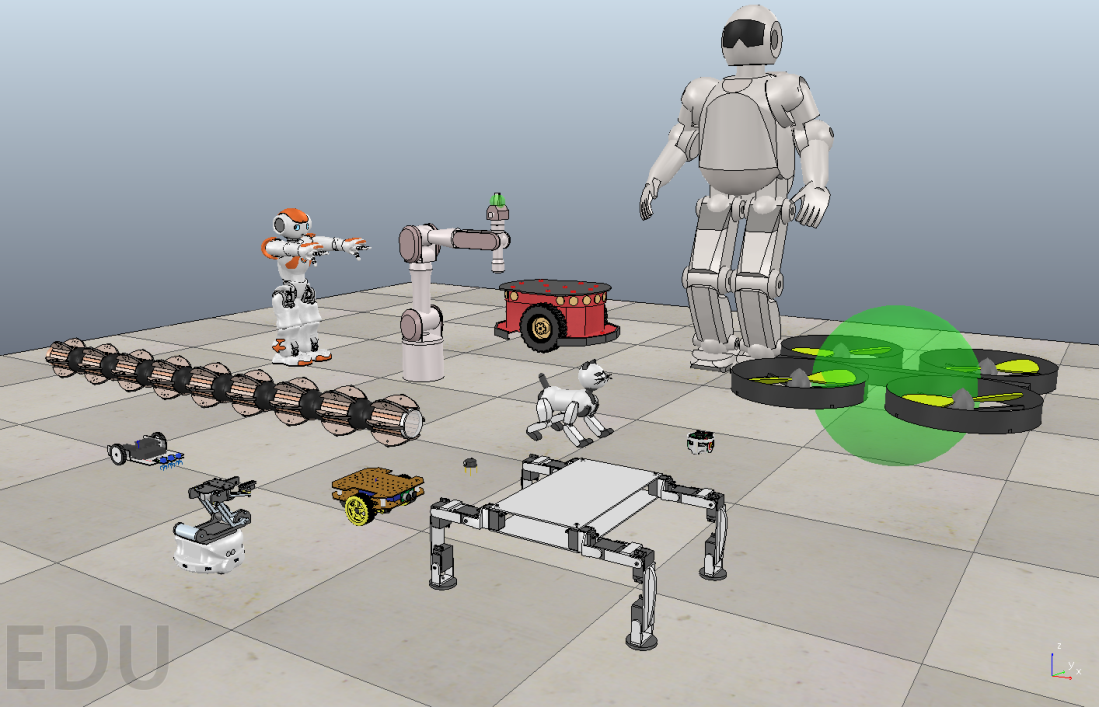


Figura X

O software V-Rep apresenta desde robôs fixos como móveis, permitindo a visualização de diversos tipos de movimentação, desde robôs com 2 graus de liberdade, até robôs homini-direcionais, trabalhando direções, obstáculos e caminhos a serem percorridos.

### MOTORES DE SIMULAÇÃO

Para a execução de cálculos e simulação de elementos de maneira realística, o software conta com quatro tipos de motores de simulação de física, isto é, ambientes programados que simulam as condições reais de interação entre corpos e materiais, considerando fatores como gravidade, atrito, peso, velocidade entre outros.

Os motores disponíveis são gratuitos e de código aberto, *Bullet Engine, ODE (Open Dynamics Engine), Vortex Engine e Newton Dynamics.* Os motores de físicas aplicados ao software, permitem que as simulações sejam pautadas em condições físicas reais, ou seja, o robô simulado pode colidir com outros corpos, sofrer deformação ou até mesmo ser destruído, fazendo com que a programação de movimento seja o mais próximo da realidade possível.

### PROGRAMAÇÃO EM LUA

O software V-Rep é bastante flexível quanto a customização das simulações criadas, apresentando mais de uma opção quanto a programação dos modelos robóticos simulados.

A maneira padrão de programação de modelos no V-Rep, é através de scripts na linguagem de programação Lua, que é uma linguagem leve e feita com a finalidade de integrar softwares escritos em diferentes linguagens, por ser tratar de uma linguagem baseada no paradigma de scripts, diferente de linguagens de estrutura compilada.

A programação em Lua é feita dentro do software V-Rep, ou integrada a simulação no momento da execução. Caso o software escrito precise ser executa externamente ao V-Rep, como por exemplo em um robô ou hardware real, é possível utilizar as funcionalidades disponibilizadas pela API do V-Rep.

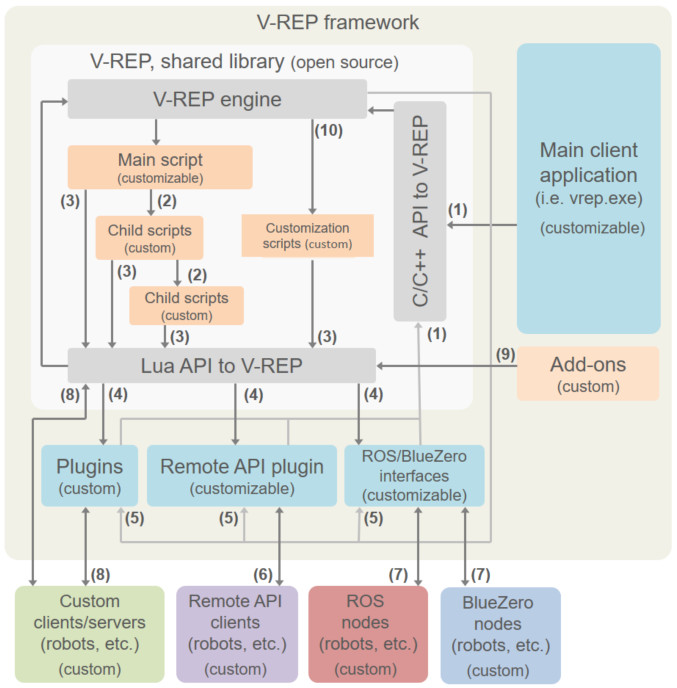
### V-REP API

Quando o software para a simulação precisa ser executado em um hardware externo ao V-Rep, é necessário utilizar a API provida pela *Coppellia Robotics*, a V-Rep API (*Application Programing Interface*), que é definida por um conjunto de funções que interagem com a simulação e podem ser integradas em diferentes plataformas e linguagens de programação.

A API do V-Rep possui suporte para interfaces de robótica, como a *ROS* e a interface *BlueZero*, utilizadas na integração de robôs com os ambientes simulados do software.

Além dessas opções, podem ser implementados clientes para a aplicação com diversas linguagens suportadas, *C/C++*, *Python*, *Matlab*, *Octave* e *Lua.* Para o desenvolvimento do programa foi desenvolvido um cliente na linguagem Python, para interagir com o código executado em hardware, no caso do projeto, em um Arduino Uno, servindo de interface entre o Hardware in The Loop e o software V-Rep.

Na FIGURA X é apresentada a arquitetura do framework do V-Rep, onde é possível observar os diferentes níveis de abstração do software, em níveis de código executado no V-Rep (scripts em Lua) e código executado em hardware alheio a simulação (API).

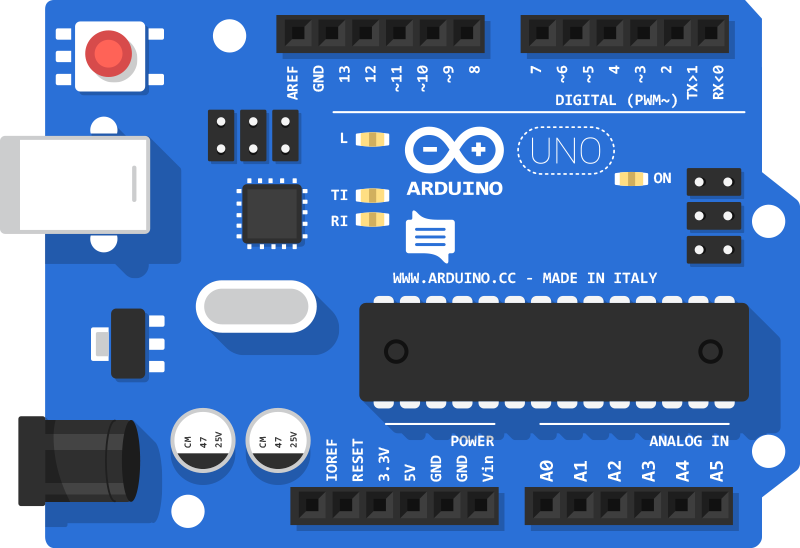


* 1. **HARDWARE DE IMPLEMENTAÇÃO E PROGRAMAÇÃO (ARDUINO)**

O Arduino é uma plataforma de prototipagem eletrônica, de hardware e software livre, que apresenta um microcontrolador *Atmel* embarcado permitindo ligamento de motores, LEDS, sensores, displays e *shields*, que aumentam as possibilidades de implementação, desenvolvimento e comunicação.

A comunidade Arduino, vem crescendo de maneira exponencial, desde sua criação, pois de maneira fácil e eficiente criam programas, componentes e situações em que o Arduino pode ser implementado, como controle de processos industriais simples, automação residencial, trabalhos com robótica, tanto móvel como industrial, controle de válvulas, etc.

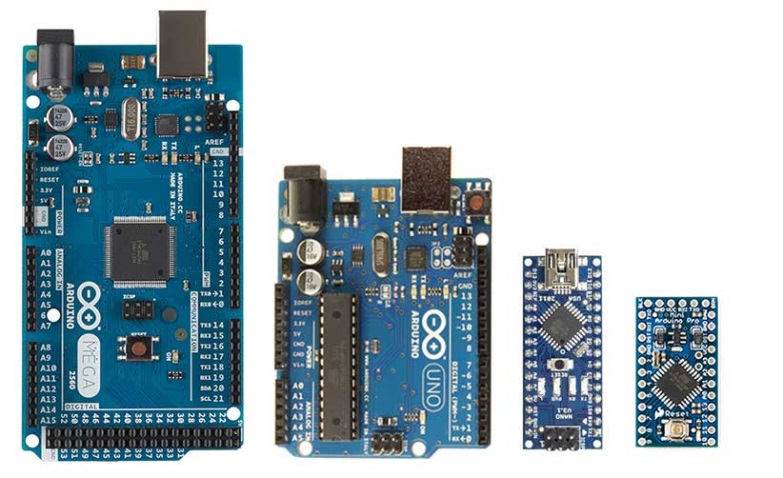
Não ficando apenas nisso ele permite controle digital e analógico, com suas I/Os múltiplas, existem várias placas de Arduino, desde as Nano, até as configurações Mega2560, todos contando com um chip-set, que podem variar sua capacidade de armazenamento interno, e processamento, por essas configurações o Arduino tem sido utilizado como o “cérebro” de vários projetos, tanto para estudantes, designers, programadores, ou até mesmo *hobbistas*. A imagem X mostra a representação de uma placa Arduino do tipo Uno.



Como é projeto de software livre, o Arduino pode ser utilizado em Mac, Windows e Linux, todos compartilhando novas bibliotecas por suas diversas comunidades de desenvolvedores. Sendo barato e preciso, podendo ser utilizado em diversos sistemas e integrando diversos recursos, rápido e eficaz, o Arduino está sendo a escolha de diversos projetos no âmbito da tecnologia por todas essas facilidades.

De hardware completo e simples temos a versão mais comum do Arduino sendo a placa UNO, outras são bem parecidas havendo apenas algumas variações de memória ROM/RAM, processamento e preço.

A imagem X abaixo apresenta alguns tipos de placa arduino lado a lado, da esquerda para a direta Arduino Mega, Arduino Uno, Arduino Nano, Arduino Mini.



## CONCEITOS DE ROBÓTICA

### 2.8.1. DEFINIÇÃO DE ROBÔ

Conforme especificado pela *Robotic Industries Association* (RIA), um robô industrial é definido como um "manipulador multifuncional reprogramável projetado para movimentar materiais, partes, ferramentas ou peças especiais, através de diversos movimentos programados, para o desempenho de uma variedade de tarefas” (RIVIN, 1988).

Já a norma ISO (International Organization for Standardization) 10218 apresenta uma definição mais completa e define um robô como sendo: "uma máquina manipuladora com vários graus de liberdade controlada automaticamente, reprogramável, multifuncional, que pode ter base fixa ou móvel para utilização em aplicações de automação industrial".

### 2.8.2. ELEMENTOS DE UM ROBÔ INDUSTRIAL

Um robô industrial, por definição, pode ser descrito como a integração de alguns elementos presentes em sua estrutura, sendo eles:

* **Manipulador mecânico**: refere-se principalmente ao aspecto mecânico e estrutural do robô, ou seja, a combinação de elementos estruturais rígidos (corpos ou elos) conectados entre si através de articulações (juntas), sendo o primeiro corpo denominado base e a última extremidade, onde será vinculado o componente efetuador (garra ou ferramenta).
* **Atuadores:** componentes que convertem energia elétrica, hidráulica ou pneumática, em potência mecânica. Através dos sistemas de transmissão a potência mecânica gerada pelos atuadores é enviada aos elos para que os mesmos se movimentem.
* **Sensores:** são os elementos que fornecem parâmetros sobre o comportamento do manipulador, em termos de posição e velocidade. As juntas utilizadas para vincular os elos de um robô são normalmente acopladas a sensores, em outras palavras, servem para localizar o robô no espaço.
* **Unidade de controle:** responsável pelo gerenciamento e monitoração dos parâmetros responsáveis pela realização de tarefas do robô. De modo simples, é o cérebro do robô, que realiza interface entre todos os elementos, realizando a rotina desejada.
* **Efetuador:** É o elemento de ligação entre o robô e o meio que o cerca. Pode ser do tipo garra ou ferramenta. A ferramenta tem como função realizar uma ação ou trabalho sobre uma peça, sem necessariamente manipulá-la.

### 2.8.3. ELEMENTOS ESTRUTURAIS

Quanto ao que diz respeito ao elemento manipulador mecânico citado na enumeração anterior, se destacam alguns conceitos importantes que devem ser observados:

* **Elos:** São os elementos rígidos dos robôs, são conectados entre si através de articulações, projetados para apresentar elevada rigidez aos esforços de flexão e torção. Comumente são empregados na construção das estruturas, alumínio e aço. Porém, recentemente têm sido usadas fibras de carbono e de vidro, além de materiais termoplásticos e plásticos reforçados.
* **Juntas:** São as articulações do robô, os elementos que unem os diferentes elos e é responsável pelo movimento do mecanismo. A disposição das juntas está diretamente ligada ao grau de liberdade de um robô, já que o número de graus de liberdade que um mecanismo robótico apresenta é o número de variáveis de posição que precisam ser especificadas para localizar o dispositivo no espaço. De modo geral, e possível dizer que o número de juntas equivale ao número de graus de liberdade.
* **Sistema de transmissão:** Os sistemas de transmissão são responsáveis por transmitir a potência mecânica dos atuadores aos elos. Para que ocorra a movimentação de cada elo do robô, é necessária a atuação de um torque ou força e velocidade ou velocidade angular gerada por um atuador. Por elementos de um sistema de transmissão, entendem-se engrenagens, correntes, correias dentadas, polias, entre outros.

A escolha dos componentes relacionados ao sistema de transmissão, e as juntas do mecanismo robótico, estão diretamente ligadas às propriedades desejadas no projeto, ou seja, velocidade de movimentação, peso suportado, precisão, fone de alimentação, de modo geral estão relacionados à eficiência mecânica e energética.

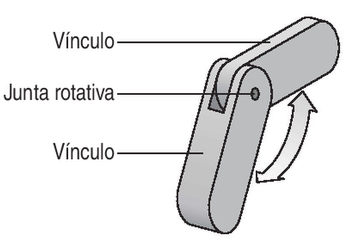
### 2.8.4. ATUADORES

Em relação aos atuadores, é possível utilizar diferentes tipos, como pneumáticos, hidráulicos ou elétricos. De modo geral, é possível definir que atuadores pneumáticos e hidráulicos proporcionam movimentos lineares, enquanto motores elétricos proporcionam moimentos angulares.

* **Atuadores pneumáticos:** Os atuadores pneumáticos, são aqueles que como o nome sugere, atuam sob efeito pneumático, ou seja, tem seu funcionamento baseado na utilização de ar comprimido. Devido sua natureza de operação binário (acionado ou não acionado), geralmente são empregados em movimentação de elementos com limitadores mecânicos. Ações de controle podem ser empregadas para se obter um controle mais delicado e suave de velocidade e posição, porém não são atuadores que proporcionam controle preciso de posicionamento nos mecanismos robóticos.
* **Atuadores hidráulicos:** Já os atuadores hidráulicos, são aqueles que atuam sob a utilização de um fluido, um tipo de óleo especifico para a aplicação. Devido à natureza de incompressibilidade do fluido hidráulico, esses atuadores proporcionam controle preciso e acurado do posicionamento dos elementos mecânicos, além de apresentar alta relação de potência mecânica transmitida, fazendo com que uma bomba pequena consiga movimentar uma quantidade considerável de carga.
* **Atuadores eletromagnéticos:** São os atuadores que tem seu funcionamento baseado em efeitos eletromagnéticos, em outras palavras, motores. São os tipos de atuadores mais utilizados na construção de mecanismos robóticos, principalmente os motores de corrente continua e motores de passo. Em conjunto com sensores, como *encoders* e potenciômetros, são os que oferecem controle mais preciso de velocidade e posicionamento dos robôs em que são utilizados, além de oferecer uma gama de fabricantes e tipos a serem utilizados em projetos.

### 2.8.5. JUNTAS

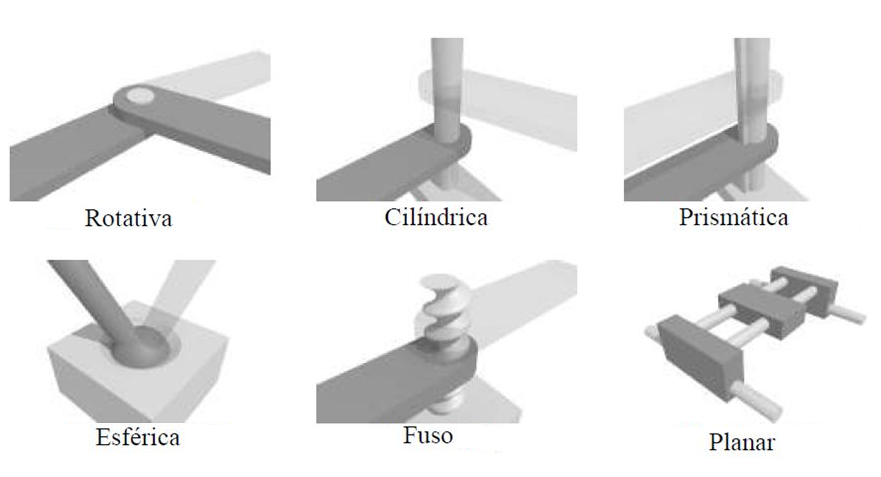
Juntas são os elementos que unem os elos do robô e proporcionam que o mecanismo robótico realize movimentos. A figura X apresenta um esquemático entre dois elos (vínculos) e uma junta do tipo rotacional.



As juntas podem ser classificadas da seguinte maneira:

* **Rotacional:** São as juntas que giram em torno de um eixo de rotação, movimentam os elos através de uma dobradiça comum as duas partes, comumente representada pela atuação de um motor elétrico;
* **Prismática, ou linear:** É o tipo de junta que proporciona movimento linear, em linha reta, geralmente representada por um pistão, ou cilindro;
* **Cilíndrica:** é composta por uma união de duas juntas, uma rotacional e uma prismática, combinando os movimentos em torno de um eixo com o movimento linear;
* **Esférica:** Esta junta funciona com a combinação de três juntas rotacionais, combinando o movimento de três eixos de rotação em uma única junta;
* **Planar:** Composta por duas juntas prismáticas, realiza movimentos em duas direções, comumente usada em configurações cartesianas;
* **Parafuso:** Junta constituída por um parafuso com uma porca, ao qual é possível realizar movimento linear análogo ao da junta prismática, porém adaptando o movimento com fundamento rotacional em um eixo central (movimento do parafuso).

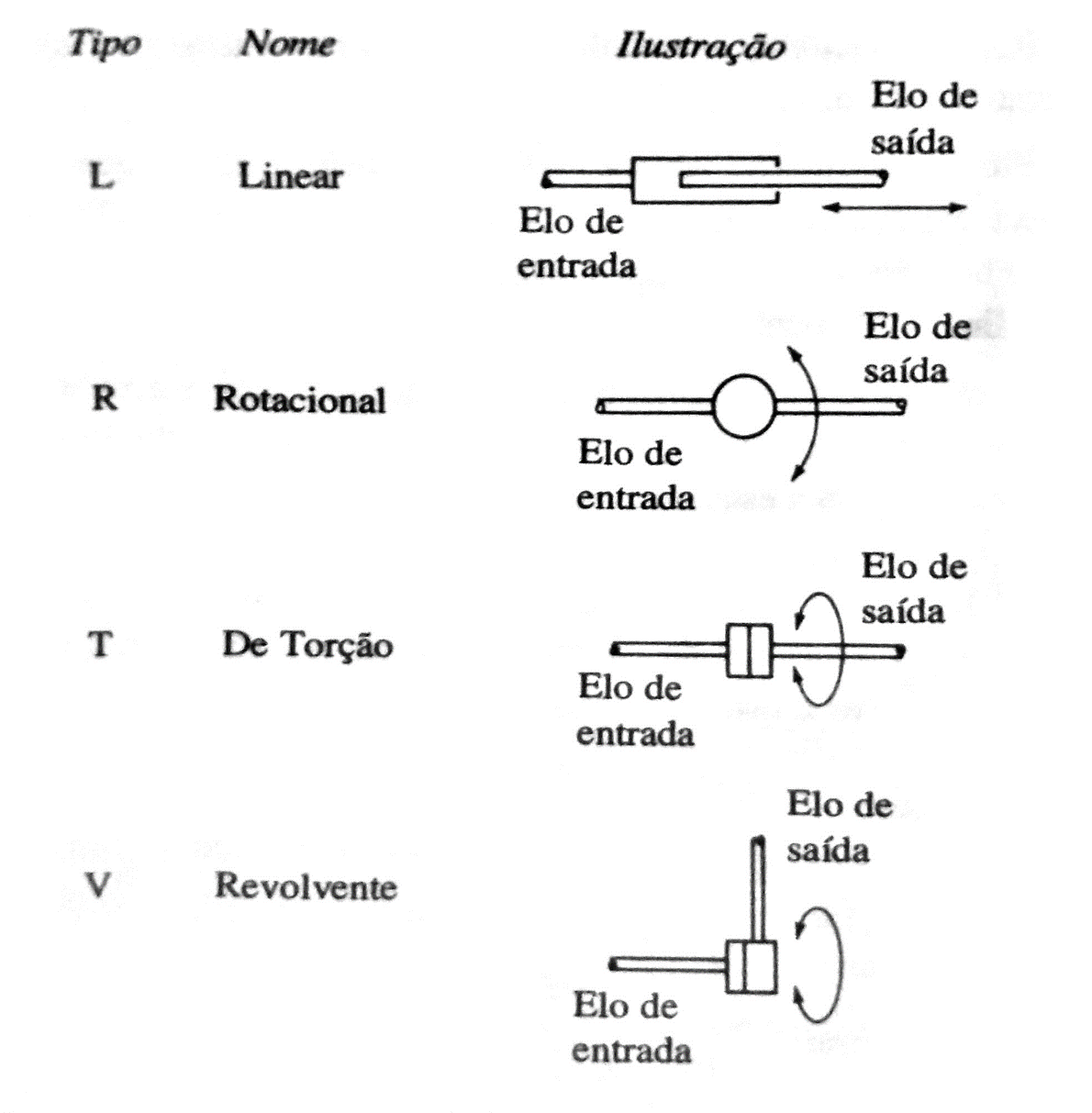
A figura X apresenta a ilustração esquemática de cada um dos tipos de juntas existentes e utilizadas em mecanismos robóticos.



No geral, robôs industriais utilizam apenas juntas prismáticas e rotativas, isso inclui a junta planar, que é considerada a junção de duas juntas prismáticas. A direção dos elos de entrada e de saída em relação ao eixo de rotação classificam as juntas rotacionais, assim, tem-se as seguintes subclassificações de juntas rotativas:

* **Linear, L:** na junta tipo L, o elo de saída mantém a direção do elo de entrada, realizando movimento linear
* **Rotacional, R:** na junta tipo R, o eixo de rotação é perpendicular aos eixos dos dois elos de conexão, realizando movimento rotacional
* **Torção, T:** a junta tipo T é uma variação da junta rotacional, onde mesmo com movimento rotacional o movimento no elo de saída mantém a direção do elo de entrada
* **Revolvente, V:** a junta do tipo V é uma terceira variação da junta R, onde o elo de entrada é paralelo ao eixo de rotação, e perpendicular ao elo de saída.

A ilustração a seguir, na figura X (GROOVER, 1988) apresenta a esquematização teórica de cada um dos tipos de juntos existentes. Os diferentes tipos de juntas são combinados a fim de construir mecanismos robóticos e classifica-los conforme o tipo de movimento que este é capaz de realizar.



Robôs industriais com frequência têm sua nomenclatura definida conforme a quantidade e tipo de juntas utilizadas. Por exemplo, um esquema simples de um robô do tipo SCARA, como apresentado na figura anterior X, poderia ser classificado como um robô RRP, uma vez que sua base e primeiro elo possuem uma junta rotacional do tipo R, enquanto a extremidade possui uma junta prismática do tipo P, com movimento vertical.

## 2.9. CLASSIFICAÇÃO DE ROBÔS

Os robôs industriais podem ser classificados de acordo com o número de juntas, o tipo de controle, o tipo de acionamento e a geometria (Engelberger, 1995). De fato, é usual classificar os robôs em relação ao seu espaço de trabalho (*workspace*), ou também em relação a estrutura de das juntas utilizadas, definindo um tipo de coordenada utilizada por cada mecanismo, isto é, como são localizadas as coordenadas no espaço pela extremidade do robô.

Segundo ROSÁRIO (????), existem cinco classes principais de manipuladores, segundo o tipo de junta, o que permite diferentes possibilidades de posicionamento no volume de trabalho. As classes ou geometrias principais de um robô, também são chamadas de sistemas geométricos coordenados, e são cartesianas, cilíndrica, esférica (ou polar), de revolução (ou articulada) e SCARA.

### 2.9.7. ROBÔ DE COORDENADAS CARTESIANAS

É classificado como robô de coordenada cartesiana, aquele que pode se movimentar em linha reta, com deslocamentos horizontais e verticais. Recebe esse nome em função do modo a localizar um ponto no espaço, através de coordenadas em um plano cartesiano (x, y e z).

Possuem geralmente três juntas deslizantes, recebendo a classificação de um robô do tipo PPP (três juntas prismáticas), e são caracterizados pela grande exatidão na localização da extremidade. A figura X ilustra o robô do tipo cartesiano.

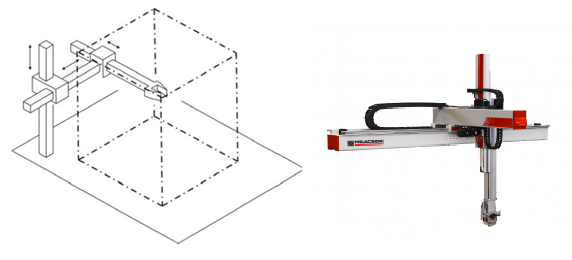


Figura X - Área de trabalho de um robô cartesiano (a) e um modelo de robô cartesiano de 3 eixos da empresa alemã Milacron (b)

### 2.9.8. ROBÔ DE COORDENADAS CILÍNDRICAS

Robôs deste tipo, combinam a ação de dois tipos de juntas diferentes, lineares e rotacionais, gerando uma área útil de trabalho em um formato cilíndrico. É codificado como RPP, ou seja, consistem em uma junta de revolução combinada com duas juntas prismáticas.

Apesar de apresentar área de trabalho ligeiramente maior que a dos robôs cartesianos, apresenta maior complexidade no controle do posicionamento, uma vez que proporciona alcance ao mesmo ponto no espaço com posicionamentos diferentes. O robô do tipo cilíndrico pode ser observado na figura X a seguir.

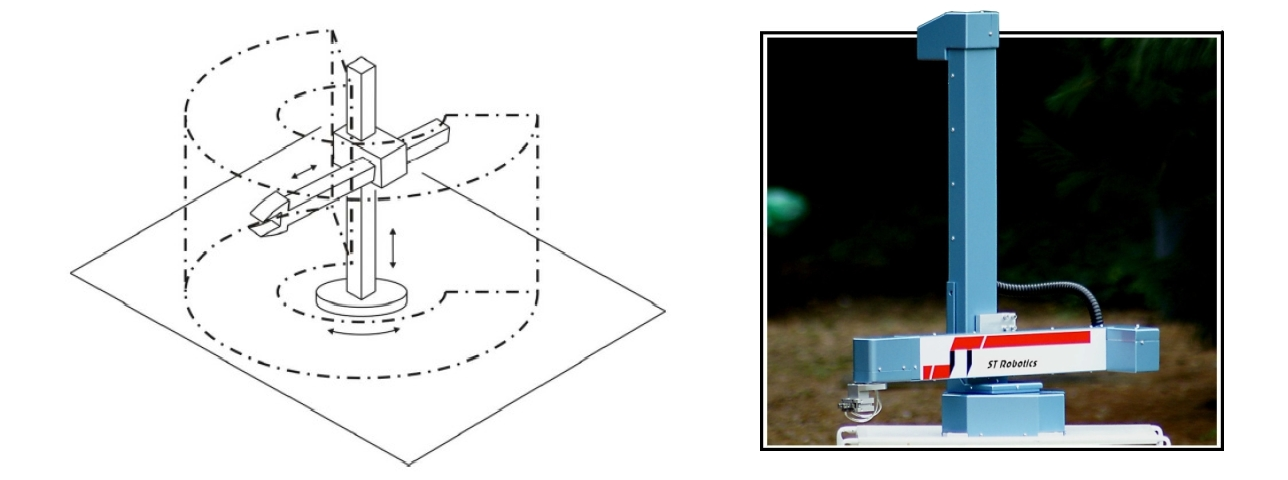


Figura X - Área de trabalho de um robô de coordenadas cilíndricas (a) e um modelo da ST Robotics do robô de coordenadas cilíndricas R19 Cylindrical

### 2.9.9. ROBÔ DE COORDENADAS POLARES (ESFÉRICAS)

Robôs de coordenadas esféricas são aqueles com três eixos que descrevem uma área de trabalho esférica. É composto por dois movimentos rotacionais, e um terceiro movimento linear, descrito assim como RRP.

Possui área de trabalho maior que os modelos cartesianos e cilíndricos, porém seu controle de posicionamento é ainda mais complexo, devido à sua área de trabalho em formato de esfera. Pode ser observado na figura X um esquema do robô de coordenadas esféricas.

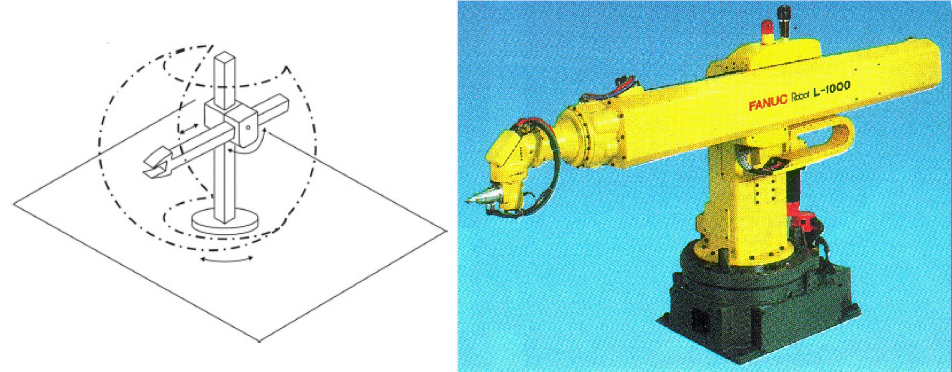


Figura X - Área de trabalho de um robô de coordenadas cilíndricas (a) e o modelo de robô cartesiano Robot L-1000 da Fanuc

### 2.9.10. ROBÔ DE COORDENADAS DE REVOLUÇÃO (ARTICULADO)

Os robôs de coordenadas de revolução, também chamados de articulados, se caracterizam por possuir três juntas de revolução, ou seja, são codificados como RRR. Com esta configuração de juntas rotacionais, o movimento de um robô articulado se assemelha ao de um braço humano.

Foi inicialmente projetado para atender as necessidades na indústria automobilística, e um dos primeiros robôs comerciais desse tipo foi o PUMA (*Programmable Universal Machine for Assembly*). A figura X mostra o esquema da área de trabalho de um robô de coordenadas de revolução.

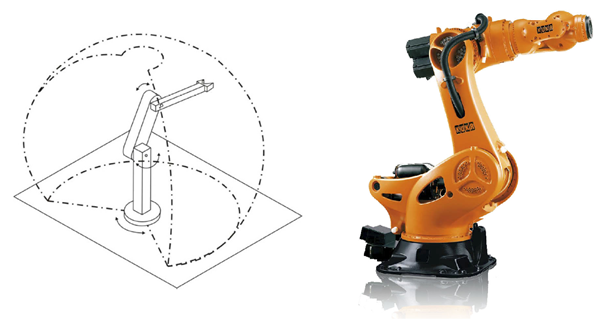


Figura X - Área de trabalho de um robô de coordenadas de revolução (a) e um robô articulado da KUKA, modelo KR 1000 Titan F

### 2.9.11. ROBÔ SCARA

O robô SCARA, é uma configuração mais recente em relação as outras utilizadas, é codificado como RRP, com duas juntas de revolução se movimentando no plano cartesiano, e uma junta prismática se movimentando no eixo z.

É utilizado principalmente em tarefas de montagem, possuindo uma área de trabalho limitada, porém grande velocidade de movimentação e precisão de localização. Pode ser confundido com um robô de coordenadas esféricas, devido a sua configuração RRP, porém apresenta área de trabalho e localização diferenciada. Na figura X é possível observar a área de trabalho de um robô SCARA.



Figura X - Área de trabalho de um robô do tipo SCARA (a) e o modelo Epson SCARA G3, da EPSON

* 1. **PROGRAMAÇÃO DE ROBÔS**

Com o crescimento do desenvolvimento dos processos de manufatura, foi necessário introduzir a ideia de que a produção fosse flexibilizada, tanto na quantidade quanto na variedade de produtos a serem produzidos.

Nesse processo de desenvolvimento surge o conceito de automação flexível, onde basicamente o layout existente no chão de fábrica, é feito de forma a flexibilizar a produção, se adaptando rapidamente a diferentes produtos ou processos.

A necessidade de rapidamente reconfigurar uma linha de produção para iniciar o desenvolvimento de outro produto ou processo, passou a exigir que as ferramentas utilizadas fossem de simples reconfiguração, surge então o conceito de *reprogramabilidade*, introduzido por Charles Devol, que desenvolveu métodos de registrar sequencias de movimentos programáveis em seus dispositivos, iniciando assim a primeira geração de robôs.

Programar um robô significa ser capaz de descrever uma sequência de movimentos e ações que se deseja que o manipulador realize a fim de desenvolver uma tarefa, utilizando para isso as funções e comandos oferecidos pelo dispositivo ao usuário.

Usualmente, existe uma interface entre o operador e o robô, com linguagens de programação legíveis a humanos que se traduzem em movimentos a serem interpretados pelo mecanismo robótico. A figura X a seguir mostra um dispositivo de interface de comunicação com um robô articulado da fabricante KUKA.



Figura

* 1. **MÉTODOS DE PROGRAMAÇÃO**

Os robôs industriais de hoje, de modo geral, são mecanismos automatizados que são projetados a fim de movimentar peças ou ferramentas sobre uma trajetória previamente definida, de modo a realizar uma tarefa desejada.

O programa do robô deverá ser capaz de se adaptar a diferentes situações desejadas, alterando funções de operação, movimentação e percurso de acordo com o processo a ser realizado. A flexibilidade de um robô é avaliada pela extensão do tipo de operações que podem ser realizadas, bem como os movimentos que podem ser programados em seu controlador.

A programação de um mecanismo robótico pode ser feita de duas maneiras: *Programação on-line*, e *Programação off-line*:

* **Programação on-line** é aquela que é realizada de fato com o robô físico e implantando em um ambiente real, e pode ser feitas utilizando técnicas de condução, aprendizagem ou linguagem textual.
* **Programação off-line** por sua vez, é realizada sem o uso do robô real, e sim com um modelo virtual simulando o ambiente final da aplicação.

A eficiência de um método de programação de um mecanismo robótico é medida levando em consideração o tempo gasto para implantar o sistema em funcionamento, no caso da programação on-line esse tempo é um parâmetro crítico, uma vez que a programação é feita com o robô real.

De modo geral, duas abordagens são utilizadas para programar a movimentação de mecanismos robóticos, quanto ao caminho a ser percorrido pelo robô: movimento ponto a ponto e movimento contínuo.

* **Movimento ponto a ponto:** o robô se move de um ponto a outro, sem considerar a trajetória desenvolvida, podendo realizar optimizações, controlando da maneira que julgar mais adequada a velocidade de cada grau de liberdade do mecanismo.
* **Movimento contínuo:** movimenta-se de um ponto a outro, mas considerando a trajetória desenvolvida, armazenando os incrementos necessários em cada eixo para percorrer o percurso de um ponto a outro.
  1. **PROGRAMAÇÃO ON-LINE**

Na programação do tipo on-line, a rotina a ser executada pelo mecanismo robótico é construída utilizando o robô real em um ambiente de aplicação ou controlado. Pode ser realizada por técnicas de aprendizagem ou utilizando linguagem textual de programação.

* + 1. **PROGRAMAÇÃO POR APRENDIZAGEM**

Quando realizada a programação por aprendizagem, o programador conduz o robô para a trajetória que deseja ser executada, de modo que o robô “aprenda” qual caminho percorrer. Este processo de “ensino” consiste em armazenar na memória do programa os pontos e instruções de operações que devem ser realizadas.

Uma vez gravados os pontos referentes à trajetória a ser percorrida e instruções a ser realizado, o robô é capaz de reproduzir os comandos inseridos e entrar em operação.

* + 1. **PROGRAMAÇÃO POR LINGUAGEM TEXTUAL**

A programação por linguagem textual se diferente da por aprendizagem pelo fato de que o robô não é conduzido até os pontos críticos de sua trajetória, todo o processo é realizado através de linguagem de programação, com comandos específicos e suportados pelo modelo do robô em que se desenvolve a rotina de operação.

Durante o processo de programação on-line, tem-se a desvantagem de que o robô fica inutilizado para operação durante a programação. O tempo em que o robô se encontra parado é um fator crítico para determinar a produtividade e flexibilidade do método de programação e operação do robô.

* 1. **PROGRAMAÇÃO OFF-LINE**

O desenvolvimento da tecnologia empregada na produção de robôs, tanto em hardware quanto em software, tem feito com que seja cada vez mais viável a utilização da programação off-line para mecanismos robóticos.

Conceitualmente, a programação off-line consiste em utilizar modelos virtuais e simulados dos robôs atuando em situações reais, utilizando a programação textual que seria de fato aplicada na instalação, e observando sua atuação em um ambiente simulado.

# **3. METODOLOGIA**

## 3.1. ESTRUTURA DO PROJETO

O desenvolvimento do modelo matemático e simulação do robô do tipo SCARA, se deu com a integração de tecnologias diferentes em ambientes diferentes. O objetivo geral, foi desenvolver um modelo matemático, capaz de descrever os movimentos do manipulador, e integra-lo em tecnologia *hardware in the loop*, utilizando software de simulação para observar visualmente as respostas do sistema implantado.

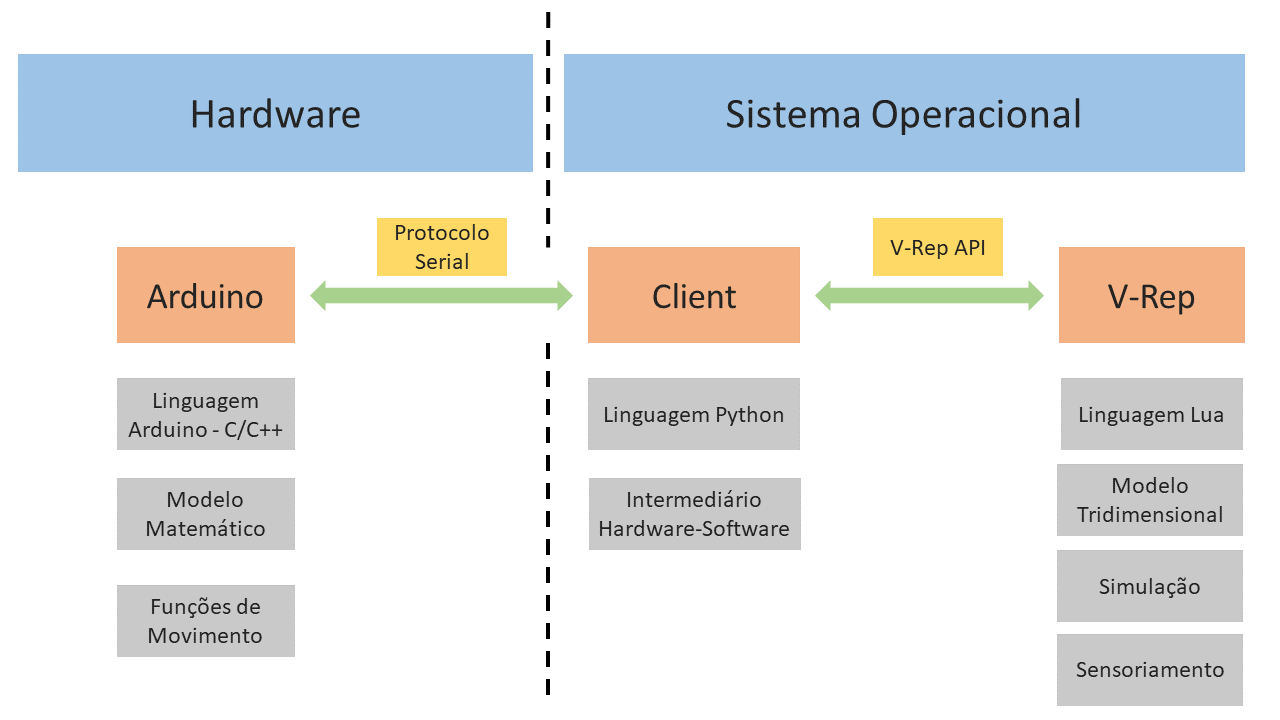
De modo geral, as etapas de desenvolvimento se dividem da seguinte forma:

1. Desenvolver modelo matemático e teórico, definindo as equações que descrevem os movimentos do manipulador;
2. Aplicar modelo matemático em uma linguagem de programação a fim de verificar a veracidade das equações desenvolvidas;
3. Definir modelo tridimensional do manipulador no ambiente de simulação a ser utilizado (V-Rep);
4. Aplicar modelo matemático em linguagem de programação embarcada no ambiente *hardware in the loop* a ser utilizado (Arduino);
5. Desenvolver comunicação entre ambiente hardware in the loop e ambiente de simulação;
6. Simular diferentes rotinas de movimentação com o manipulador robótico.

Para a etapa de aplicação do modelo matemático em linguagem de programação, a fim de verificar a veracidade das equações desenvolvidas, foi utilizado a linguagem Python, e para a linguagem de programação no ambiente embarcado, foi utilizado a própria linguagem da plataforma escolhida, o Arduino, que é uma linguagem baseada em C/C++.

A comunicação entre o ambiente hardware in the loop e o ambiente de simulação, se deu através do protocolo de comunicação *serial* e todas as simulações e etapas foram realizadas em um computador utilizando sistema operacional *Windows.* Para o ambiente de simulação, foram aplicadas rotinas programadas na linguagem *Lua*, que é a linguagem padrão do software V-Rep.

A comunicação com o ambiente simulado no software V-Rep, precisa ser feita através das funções da API disponibilizada. Para isso, foi escrito um *client* em P*ython*, ou seja, um programa que atua como intermediário na comunicação entre o *hardware* e a simulação. O esquema geral da estrutura do projeto pode ser observado na figura X abaixo.

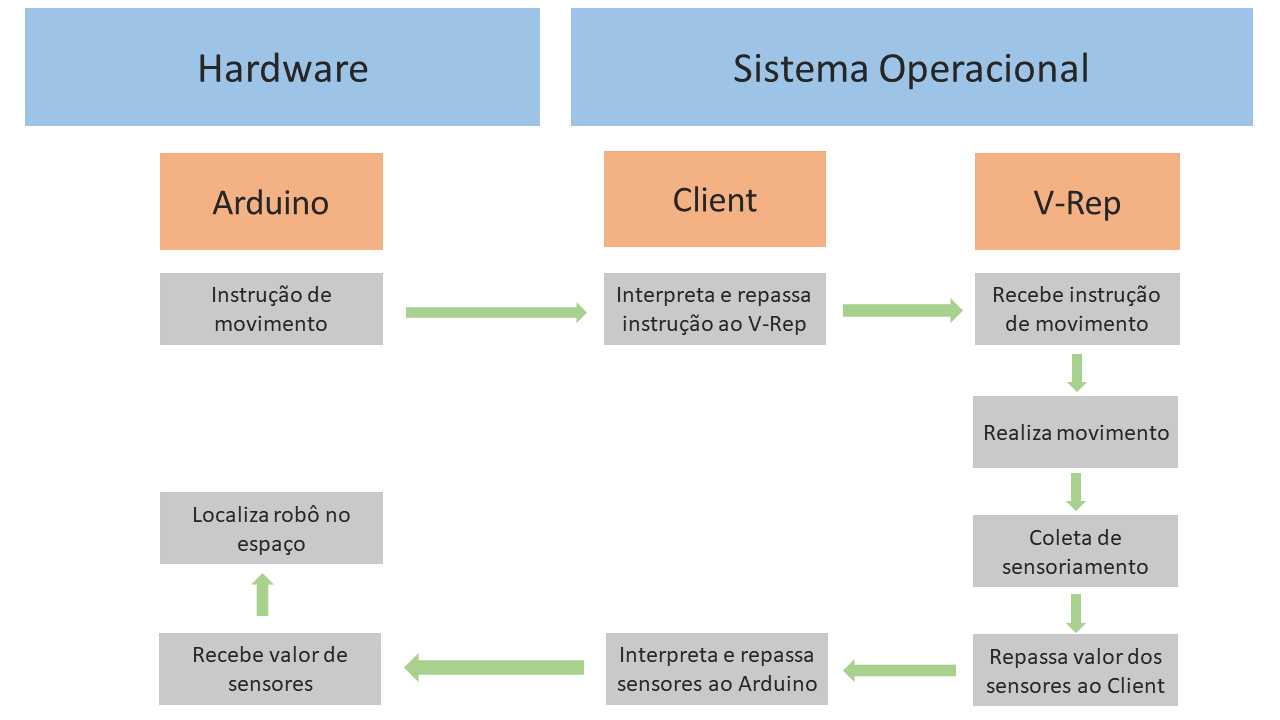


Figura

O projeto divide-se em basicamente duas partes, hardware e software, sendo a primeira parte a plataforma embarcada (Arduino) e a segunda o sistema operacional utilizado (Windows), contendo o *client* e o software de simulação V-Rep.

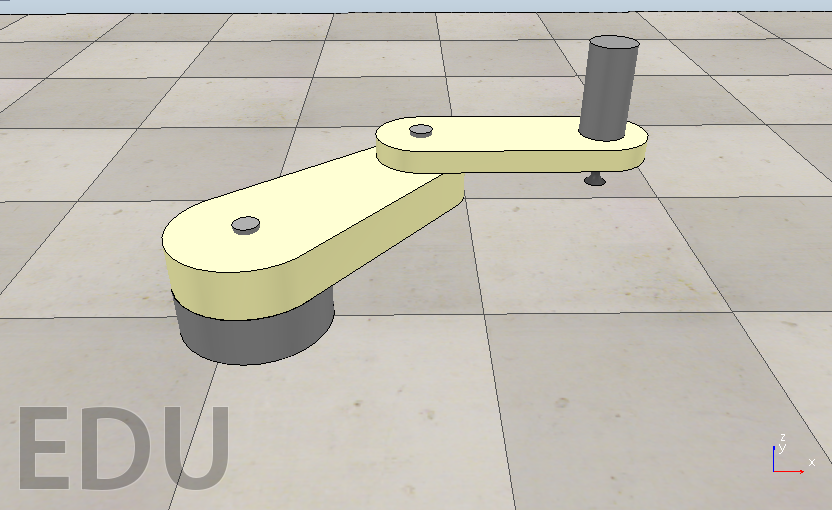
No Arduino ficam os códigos responsáveis por descrever o modelo matemático e as funções de movimento do robô. No *client* ficam as rotinas responsáveis por se comunicar com o Arduino via protocolo Serial e repassar informações para o simulador através da V-Rep API.

O *client* tem também a função de repassar as informações de sensoriamento do modelo simulado no V-Rep para o Arduino. Em um ambiente real, os sensores instalados no robô seriam responsáveis por dizer ao controlador onde o mesmo se encontra no espaço, já em um ambiente simulado, são utilizadas as leituras de sensoriamento do próprio software de simulação.



A figura X acima apresenta de maneira geral a rotina realizada durante a simulação do robô. Os valores de sensores colhidos pelo V-Rep são enviados ao Arduino para representar a leitura de um sensor real, o robô então se localiza no espaço e dá procedimento a rotina de movimentação.

Para a simulação, foi utilizado um modelo tridimensional de um manipulador SCARA disponibilizado no software V-Rep. O modelo conta com as duas juntas rotacionais dos elos 1 e 2, e com uma junta prismática na extremidade, com uma ferramenta do tipo sucção, como uma ventosa. O robô utilizado é apresentado na figura X a seguir.



Durante a simulação, o robô pode interagir com outros elementos tridimensionais, simulando colisões e a própria ação de pegar ou soltar um objeto. Assim, será possível implementar rotinas de montagens ou movimentação de peças para demonstrar a precisão e funcionalidade do modelo desenvolvido.

Para seguir com o desenvolvimento do projeto, a primeira etapa é realizar a modelagem matemática do sistema, seguindo da verificação do modelo com uma linguagem de programação.

**3.2. MODELAGEM MATEMÁTICA**

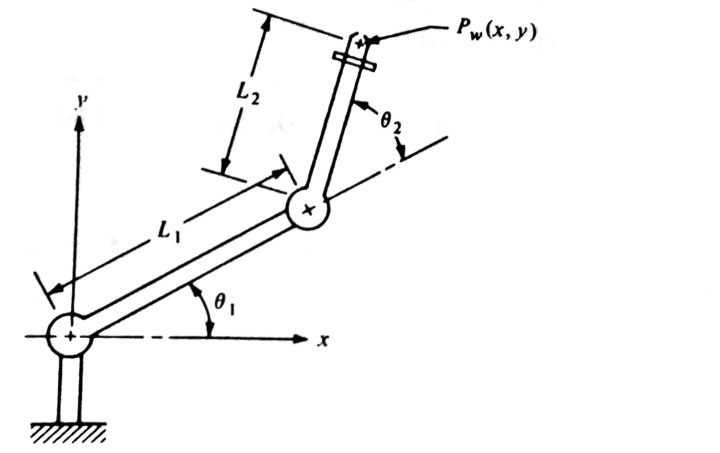
Para desenvolver um método de controlar o movimento de um manipulador robótico, é necessário descrever o movimento do robô matematicamente, isto é, definir quais equações descreve o movimento a ser realizado, localizando o robô no espaço.

As variáveis que irão descrever o movimento de um mecanismo robótico, em específico o manipulador SCARA, de maneira geral, englobam comprimento dos elos, ângulos de rotação e coordenadas desejadas. Para definir o modelo matemático do manipulador, foram feitas duas abordagens, um modelo matricial e um modelo trigonométrico.

### 3.2.1. MODELAGEM MATRICIAL

Como citado anteriormente, o movimento do robô poder feito utilizando das técnicas de cinemática, que incluem cinemática inversa e cinemática direta, realizando a passagem do espaço cartesiano para o espaço das juntas e do espaço das juntas para o espaço cartesiano, respectivamente.

Para desenvolver os modelos matemáticos, foi tomada como esquema a figura X, que ilustra as nomenclaturas de ângulos e comprimento de elos do manipulador SCARA.



Desse modo, o ângulo da junta rotacional da base é referido como , enquanto o da junta rotacional entre os elos é referido como . Já os comprimentos dos elos são tratados como e . Com essas informações, é possível definir os elementos de Denavit-Hartenberg, como observado na tabela X a seguir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Ligamento* |  |  |  |  |
| 1 | l1 | 0 | 0 |  |
| 2 | l2 | 0 | 0 |  |

É possível observar que cada elo do manipulador SCARA aponta para uma posição no espaço, sendo assim, considera-se cada elo do robô como sendo um vetor, que aponta para uma posição com uma direção , observa-se também que de tal modo, o vetor correspondente ao elo 2 tem como referência o vetor do elo 1.

Representando cada vetor, ou seja, cada elo, como uma matriz de referência, é possível localizar o elo no espaço, indicando sua origem e sua posição final. Para fins de cálculo de projeto, a origem do sistema foi considerada no inicio do elo 1.

Os elementos contidos nas matrizes são definidos utilizando propriedades trigonométricas para localização dos vetores. Sendo assim, as expressões X e X indica as matrizes de referência que representam os elos 1 e 2, representados como matrizes A e B, respectivamente.

|  |  |  |
| --- | --- | --- |
|  |  | X |
|  |  |  |
|  |  | X |

Utilizando a definição de que o vetor correspondente ao elo 2 inicia-se no fim do elo 1, ou seja, tem como referência o elo 1, a operação vetorial que representa um terceiro vetor correspondente a posição final do sistema, é definido como a multiplicação das matrizes que representam os dois vetores.

A multiplicação vetorial refletida nas matrizes gera uma terceira matriz, contendo a posição final do atuador, que nada mais é do que a indicação de um terceiro vetor correspondente à posição da extremidade do atuador. A matriz contendo a multiplicação é observada na expressão X a seguir.

|  |  |
| --- | --- |
|  | X |

Sabendo que a matriz corresponde a um vetor resultante construído a partir da multiplicação dos vetores dos elos e representa a posição final do atuador, as coordenadas finais do atuador estão contidas nesta mesma matriz, mais precisamente na quarta coluna da matriz. Deste modo, é possível resumir a transformação do braço com a cinemática inversa utilizando estes termos, observados nas expressões X e X.

|  |  |  |
| --- | --- | --- |
|  |  | X |
|  |  |  |
|  |  | X |

Computacionalmente, executar funções para calcular a formula final para um modelo generalizado é melhor do que recalcular a multiplicação matricial a cada instante que se deseja realizar um movimento, portanto, as fórmulas acima são as que serão implementadas nos modelos programados.

* + 1. **MODELAGEM TRIGONOMÉTRICA**

Além da modelagem matricial, é possível desenvolver pra alguns modelos de robôs, um modelo trigonométrico. O robô do tipo SCARA especialmente apresenta um modelo trigonométrico simples de ser desenvolvido.

Tomando novamente o esquema da figura X como representação do manipulador utilizado, foram utilizadas as mesmas nomenclaturas para descrever modelagem do manipulador.

De modo geral, o modelo trigonométrico objetiva demonstrar o posicionamento do manipulador SCARA espacialmente com propriedades trigonométricas, traçando um triângulo entre a origem do robô, os elos e a extremidade, conforme observado na figura X a seguir.



Assim, a modelagem trigonométrica para realizar a cinemática inversa, visa relacionar matematicamente através de propriedades trigonométricas a posição final do atuador (*x, y*) em função dos ângulos aplicados nas juntas ( e ), de modo a resolver todas as equações sem a utilização de matrizes de referencia como no modelo anterior.

Para desenvolver a cinemática direta do robô SCARA, isto é, descobrir as coordenadas da extremidade do manipulador em função dos ângulos conhecidos nas juntas, novamente é utilizado o conceito de elos representados por vetores, sendo assim, podemos definir que a posição final do braço no espaço cartesiano é definida por dois “vetores”, um para o elo 1 e outro para o elo 2.

|  |  |  |
| --- | --- | --- |
|  |  | X |
|  |  |  |
|  |  | X |
|  |  |  |
|  |  | X |

Novamente, a soma vetorial dos vetores gera as coordenadas *x* e *y* da extremidade do braço para o (ponto ) no espaço cartesiano. Nota-se que expressão X resultante, é a mesma resultante do processo matricial.

|  |  |  |
| --- | --- | --- |
|  |  | X |
|  |  |  |
|  |  | X |
|  |  |  |

Extraindo as coordenadas x e y do vetor resultante v3, se obtém a expressão que relaciona a coordenada do manipulador em função dos valores conhecidos dos ângulos aplicado nas juntas.

|  |  |  |
| --- | --- | --- |
|  |  | X |
|  |  |  |
|  |  | X |

As expressões X e X apresentam a posição do manipulador no espaço, partindo do pressuposto de que os ângulos aplicados nas juntas são conhecidos, ou seja, realiza a cinemática direta. Para realizar a cinemática inversa, é preciso desenvolver matematicamente a expressão para colocar as equações em função das coordenadas *x* e *y*.

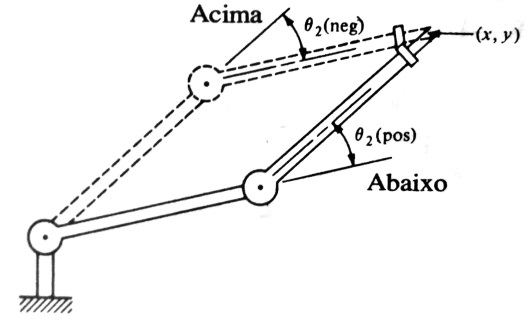
A fim de desenvolver as expressões, foram utilizadas propriedades trigonométricas para desenvolver o cálculo das funções cosseno e seno de uma soma de ângulos.

|  |  |  |
| --- | --- | --- |
|  |  | X |
|  |  |  |
|  |  | X |

Utilizando as propriedades, é possível reescrever as expressões X e X como:

|  |  |  |
| --- | --- | --- |
|  |  | X |
|  |  |  |
|  |  | X |

No manipulador do tipo SCARA, cada posição no espaço pode ser acessada utilizando duas configurações possíveis de juntas, como observado na figura X, isso faz com que existam dois valores possíveis para o ângulo .



Para fins de generalização da fórmula, as expressões a seguir considera o ângulo na segunda junta como positivo. Continuando o desenvolvimento das equações, elevando ambas ao quadrado e somando as duas, obtém-se:

|  |  |  |
| --- | --- | --- |
|  |  | X |

Conforme observado na figura X, é necessário definir os valores dos ângulos α e β a fim de continuar com o cálculos dos ângulos e .

|  |  |  |
| --- | --- | --- |
|  |  | X |
|  |  |  |
|  |  | X |

Nesta etapa, utiliza-se mais uma identidade trigonométrica para resolver a tangente do ângulo para obter a equação X.

|  |  |  |
| --- | --- | --- |
|  |  | X |
|  |  |  |
|  |  | X |

Agora, conhecendo os comprimentos dos elos l1 e l2, pode-se calcular os ângulos requeridos das juntas para colocar o braço numa posição (*x*, *y*) no espaço cartesiano. As equações X e X são as que serão implementadas utilizando linguagem de programação para realizar o movimento do braço robótico.

* 1. **MODELADEM EM PYTHON**

Após definir as fórmulas a serem utilizadas conforme o modelo matemático demonstrado, a etapa seguinte foi o desenvolvimento de um primeiro modelo programado na linguagem Python.

Este modelo foi utilizado para colocar o conceito em prova e verificar a confiabilidade das equações obtidas, utilizando de gráficos e recursos visuais para demonstrar a utilização das equações de cinemática inversa e direta.

Vale ressaltar, que o modelo em python neste momento ainda não apresenta ligação com a simulação do robô, e sim apenas com a visualização prévia das funções de cinemática e visualização do *workspace* do robô a ser desenvolvido.

* + 1. **BIBLIOTECAS E DEPENDÊNCIAS**

Durante o desenvolvimento do modelo cinemático em python, foram utilizadas algumas bibliotecas para auxiliar com funções principalmente matemáticas e de visualização gráfica.

Para realizar a plotagem dos gráficos, foi utilizada a biblioteca Matplotlib, e para funções matemáticas foi utilizada a biblioteca científica Numpy. A figura X mostra o cabeçalho com as bibliotecas utilizadas.

1. # -\*- coding: utf-8 -\*-
2. **import** matplotlib.pyplot as plt
3. **import** numpy as np
4. **from** numpy **import** cos, sin, arccos, arctan2, power
5. **from** numpy **import** radians, degrees, array

Todas as bibliotecas utilizadas, bem como a linguagem python, são de acesso gratuito e podem ser adquiridas através dos sites oficiais, e contem ampla documentação e comunidade ativa, disponíveis para todas as plataformas Windows, Linux e Mac.

### 3.2.1. CINEMÁTICA DIRETA EM PYTHON

Após as fórmulas serem definidas, foi possível programar funções para realizar as transformações cinemáticas conforme os parâmetros fornecidos. Foram programadas funções para realizar a cinemática inversa e direta do robô.

A figura X abaixo mostra o código para a função de cinemática direta. Nesta função, é realizada a passagem do espaço das juntas para o espaço cartesiano, utilizando as equações demonstradas anteriormente (equação X).

1. #Transformação Direta
2. #Passagem do Espaço das Juntas p/ o Espaço Cartesiano
3. **def** direct\_transform(t1, t2, l1, l2):
5. #Computa vetores individuais para elos L1 e L2
6. r1 = array([l1 \* cos(radians(t1)), l1 \* sin(radians(t1))])
7. r2 = array([l2 \* cos(radians(t1 + t2)), l2 \* sin(radians(t1 + t2))])
9. #Soma os vetores individuais p/ obter coordenadas do ponto P(x)
10. r3 = array(r1 + r2)
12. #Retorna vetores individuais r1, r2 e vetor final r3
13. **return** [r1, r2, r3]

Nesta função, são recebidos como parâmetros os ângulos nas juntas rotacionais *t1* e *t2* em graus*,* bem como os comprimentos dos elos *l1 e l2* em metros, a resposta em forma de um vetor de três posições que contém a posição calculada dos elos individuais *r1* e *r2*, e a posição final do atuador *r3*, ambas em coordenadas cartesianas *x* e *y*.

É possível notar que no código há conversões realizadas entre graus e radianos. As fórmulas demonstradas utilizam todas as unidades de angulo como graus, mas computacionalmente a maioria das funções matemáticas utilizam as unidades de angulo como radianos, assim é necessário adaptar os valores utilizados durante a função.

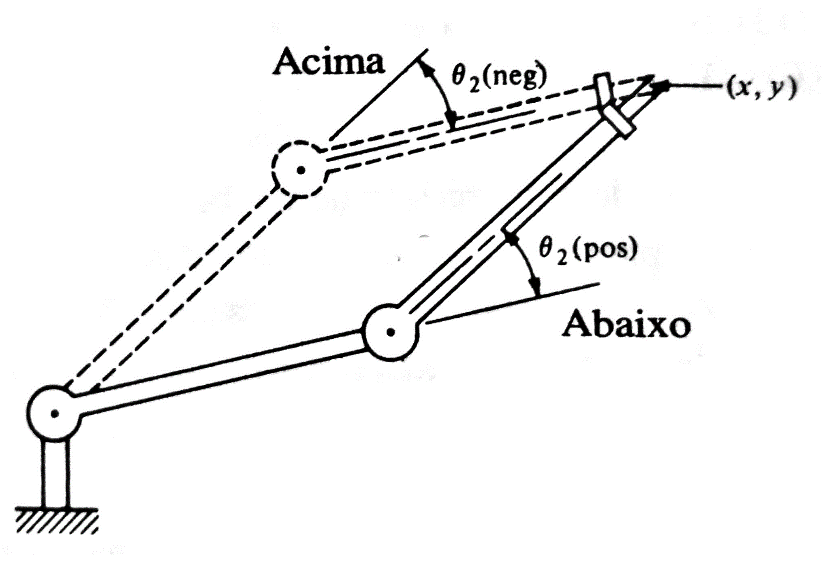
* + 1. **CINEMÁTICA INVERSA EM PYTHON**

Na cinemática inversa, é realizada a passagem contrária à cinemática direta, ou seja, é feita a passagem do espaço cartesiano para o espaço das juntas. As equações que fundamentam esta passagem encontram-se demonstradas em equação X.

A figura X apresenta a função de cinemática inversa, no qual são recebidos como parâmetros às coordenadas do ponto cartesiano *x* e *y*, e os comprimentos dos elos *l1* e *l2*.

1. #Transformação Inversa
2. #Passagem do Espaço Cartesiano para o Espaço das Juntas
3. **def** inverse\_transform(x, y, l1, l2):
5. #Vetor para armazenar os resultados finais
6. res = []
8. #Inicia cálculos para multiplicador -1 e 1
9. **for** i **in** (-1, 1):
11. #Computa valor de t2
12. t2=arccos((power(x,2)+power(y,2)-power(l1,2)-power(l2,2))/(2\*l1\*l2))
14. #Atualiza multiplicador de t2
15. t2 \*= i
17. #Verifica se t2 retornou NaN
18. **if** np.isnan(t2): t2 = 0
20. #Computa valor de t1
21. num = y \* (l1 + l2 \* cos(t2)) - x \* l2 \* sin(t2)
22. den = x \* (l1 + l2 \* cos(t2)) + y \* l2 \* sin(t2)
23. t1 = arctan2(num, den)
25. #Converte em graus
26. t1 = degrees(t1)
27. t2 = degrees(t2)
29. #Adiciona resposta ao vetor de saída
30. res.append(t1)
31. res.append(t2)
33. #Retorna os ângulos possíveis
34. **return** res

É importante notar que para a configuração de um manipulador do tipo SCARA, existem duas maneiras de alcançar um determinado ponto dentro da área de trabalho do robô. A figura X ilustra a situação, demonstrando configurações diferentes das juntas que levam a extremidade do braço para a mesma posição.



Assim, foi utilizado no código um multiplicador, para realizar os dois cálculos quando solicitado a função de transformação inversa, o multiplicador alterna seu valor de -1 para 1, calcula as duas possíveis configurações de acesso ao ponto no espaço cartesiano.

Deste modo, a função retorna um vetor de 4 posições, onde cada par de valores representa uma configuração de juntas *t1* e *t2* para acesso ao ponto no espaço dentro da área de trabalho.

* + 1. **TESTANDO O MODELO PROGRAMADO**

Com as funções de cinemática inversa e direta programadas, é possível colocar o código em teste, inicialmente foram realizados testes apenas numéricos, e em seguida testes utilizando recursos visuais.

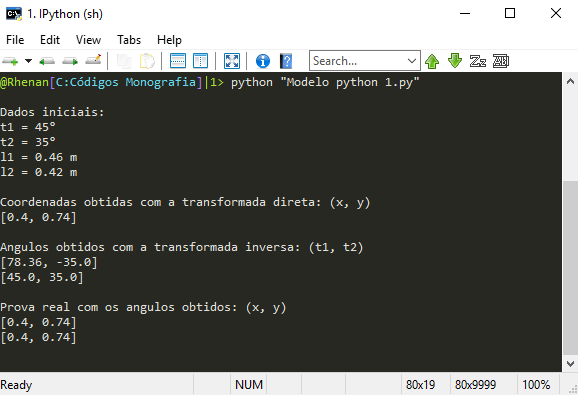
A função de transformada direta realiza a passagem de uma configuração de juntas para uma posição no espaço cartesiano, e a transformada inversa realiza a passagem contrária, de um ponto no espaço para as configurações de juntas possíveis.

Logo, em teoria, ao utilizar a resposta de uma transformação direta como entrada para uma transformação inversa, o resultado obtido deve ser igual ao parâmetro utilizado como entrada no início do processo. A figura X abaixo, demonstra o trecho de código utilizado para realizar a “prova real” entre as duas transformações.

1. #Parâmetros de teste
3. #Conjunto de ângulos iniciais
4. t1 = 45
5. t2 = 35
7. #Comprimento dos elos l1 e l2
8. l1 = 0.460
9. l2 = 0.400
11. #Computa transformação direta com ângulos iniciais
12. direct = direct\_transform(t1, t2, l1, l2)
14. #Computa transformação inversa partindo das coordenadas obtidas
15. inverse = inverse\_transform(direct[2][0], direct[2][1], l1, l2)
17. #Gera prova real para os ângulos obtidos
18. direct1 = direct\_transform(inverse[0], inverse[1], l1, l2)
19. direct2 = direct\_transform(inverse[2], inverse[3], l1, l2)

Os valores utilizados como ângulos iniciais em *t1* e *t2* são números arbitrários, e podem ser qualquer valor dentro da extensão possível das juntas utilizadas. Para testes, foram consideradas juntas com alcance de 0º a 360º. Os comprimentos dos elos l*1* e *l2* foram igualmente atribuídos de forma arbitrária, representando elos com comprimentos de 46 cm e 42 cm.

Executando a rotina de testes, os resultados obtidos são visualizados na figura X, que mostra o *console* em python após a execução do *script.* Durante os testes com o modelo programado, foi utilizado como saída para os resultados números o console IPython.



Figura

Para comprovar a veracidade do modelo desenvolvido, serão analisados os três conjuntos de números apresentados como saída:

* **Coordenadas obtidas com a transformada direta:** coordenadas obtidas com a transformação direta, realizada com os dados iniciais. Valores apresentados em coordenadas cartesianas *x* e *y;*
* **Ângulos obtidos com a transformada inversa:** ângulos obtidos com a transformação inversa, utilizando as coordenadas anteriores como entrada. Valores apresentados em graus;
* **Prova real com os ângulos obtidos:** os ângulos calculados serão utilizados para retornar para o espaço cartesiano, verificando se o modelo está correto ou não. Valores apresentados em coordenadas cartesianas *x* e *y;*

Os passos realizados pela rotina de testes, com os resultados visualizados na saída do console IPython se dão da seguinte maneira:

1. Com os parâmetros iniciais é realizada a transformação direta, onde são calculadas as coordenadas de onde estará a extremidade do manipulador. São apresentadas como saída, as coordenadas finais *x* e *y*;
2. Após obter as coordenadas *x* e *y*, elas são utilizadas como parâmetro para realização da transformação inversa, nesta etapa, são apresentados dois conjuntos de ângulos para *t1* e *t2*, lembrando que cada ponto no espaço pode ser acessado de duas maneiras diferentes. Observa-se, que um dos conjuntos calculados, é o mesmo utilizado como entrada, mostrando que o modelo segue realizando os cálculos de modo correto.
3. Com o par de ângulos possíveis sendo conhecidos, é realizada a transformação inversa com esses valores. Com um modelo programado corretamente, era de se esperar que as coordenadas apresentadas fossem iguais as obtidas inicialmente, foi possível constatar essa igualdade como conferida no terceiro conjunto de saídas da rotina.
   * 1. **VISUALIZAÇÃO DA ÁREA DE TRABALHO**

Comprovando numericamente a veracidade do modelo programado, a próxima etapa é comprovar visualmente o funcionamento do modelo, para isso, foi utilizada a biblioteca *Matplotlib* para primeiro *plotar* a área de trabalho do robô, e depois visualizar posições do robô conforme os valores inseridos de juntas e coordenadas.

Para gerar a área de trabalho, ou *workspace* do manipulador, foi utilizado um método em que é realizada a transformação direta para cada uma das configurações possíveis de ângulos nas juntas *t1* e *t2*, então as coordenadas do ponto obtido foram armazenadas em vetores para os eixos *x* e *y*, e por fim, os eixos foram plotados em um gráfico de coordenadas cartesianas.

A primeira etapa, conforme apresentada na figura X é definir os parâmetros do manipulador para gerar o workspace, definindo tamanho dos elos e ângulos máximo e mínimo para as juntas, bem como os vetores para armazenar as coordenadas dos pontos x e y. Para o modelo em teste, foi utilizado uma extensão de -25° a 205° para a primeira junta rotacional, e -115° a 115° para a segunda junta rotacional.

1. #Array para armazenar os pontos, em x e y
2. pontos\_x = []
3. pontos\_y = []
5. #Comprimento dos elos l1 e l2
6. l1 = 0.460
7. l2 = 0.350
9. #Valores máximo e mínimo para ângulo na junta 1
10. t1\_min = -25
11. t1\_max = 205
13. #Valores máximo e mínimo para ângulo na junta 2
14. t2\_min = -115
15. t2\_max = 115

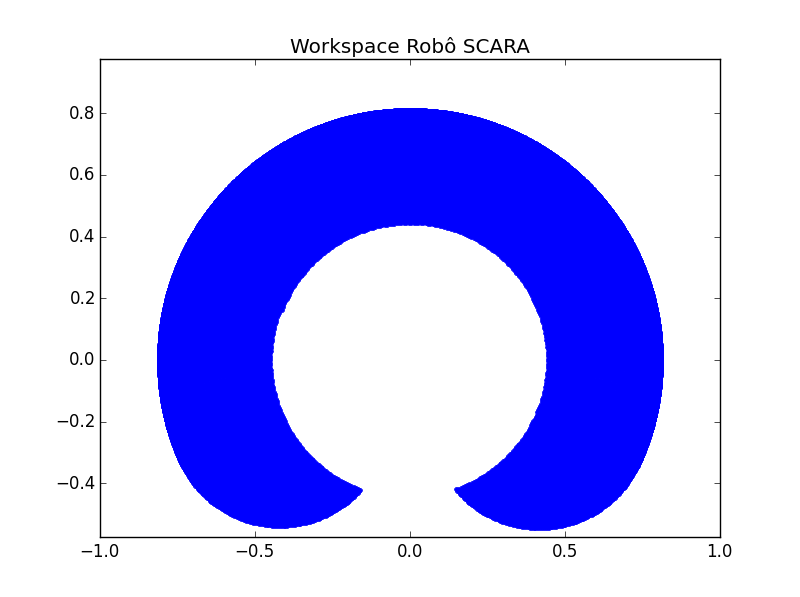
Em seguida, são gerados os pontos em coordenadas cartesianas para cada uma das combinações possíveis dentro do alcance das juntas. Para isso, foi utilizado dois *loops* aninhados que percorrem cada uma dessas combinações, realiza a transformação direta e armazena as coordenadas de resposta nos vetores definidos. Este processo é observado na figura X a seguir.

1. #Gera Workspace
2. **for** i **in** range(t1\_min, t1\_max):
3. **for** j **in** range(t2\_min, t2\_max):
5. #Computa transformação direta
6. direct = direct\_transform(i, j, l1, l2)
8. #Adiciona o ponto ao vetor de coordenadas
9. pontos\_x.append(direct[2][0])
10. pontos\_y.append(direct[2][1])

Por fim, é necessário visualizar graficamente, realizando a plotagem de um gráfico em um eixo de coordenadas cartesianas com os pontos armazenados nos vetores de posição em *x* e *y*. A figura X a seguir apresenta o código que realiza a plotagem do gráfico.

1. #Plota Workspace
2. plt.figure()
3. plt.title(u"Workspace Robô SCARA")
4. plt.axis('equal')
5. plt.plot(pontos\_x, pontos\_y, marker='.', linestyle='none')
6. plt.show()

Foi utilizada a função de plotagem da biblioteca Matplotlib, obtendo o gráfico apresentado na figura X a seguir, que representa a área de trabalho do manipulador SCARA, onde os valores representados nos eixos estão em unidade de metros.

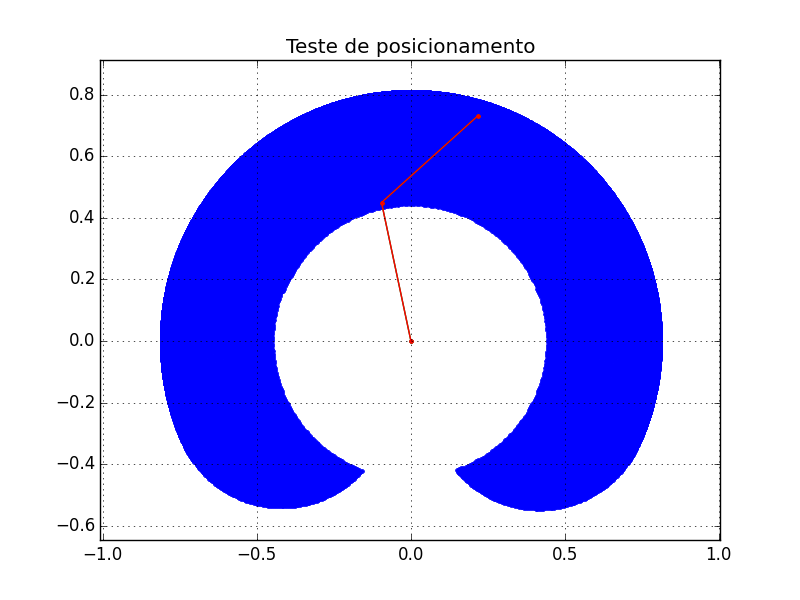


* + 1. **TESTE DE POSICIONAMENTO**

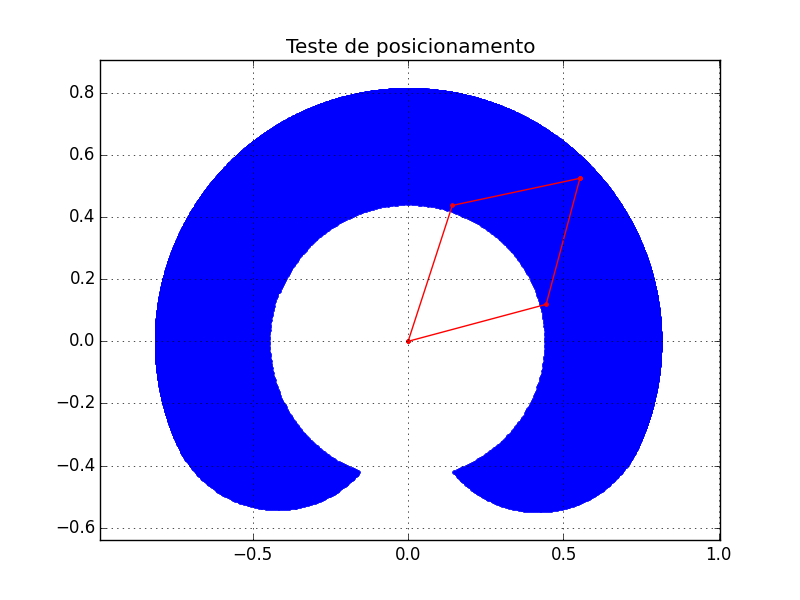
Ao realizar os testes para verificar o funcionamento do modelo matemático, todas as saídas eram apenas visualizadas numericamente, a etapa seguinte foi desenvolver um método de visualizar graficamente os resultados apresentados.

Para esta tarefa, foram utilizados os recursos gráficos da linguagem Python, utilizando da biblioteca Matplotlib, conforme apresentada na apresentação do workspace do robô SCARA.

Conhecendo os vetores que representam os elos l1 e l2, é fácil demonstra-los graficamente, basta traçar uma linha que represente o vetor. A origem para o vetor do elo 1, é o ponto cartesiano (0, 0), já a origem para o vetor do elo 2, é o fim do elo 1. A figura X apresenta um exemplo de teste de posicionamento do modelo desenvolvido em python.



Com o teste de posicionamento, é possível visualizar a configuração dos elos do braço robótico, bem como a posição final do atuador e comparar o resultado visto graficamente com os apresentados numericamente. É possível também, demonstrar a característica do manipulador SCARA de oferecer duas configurações possíveis de acesso ao mesmo ponto no espaço.

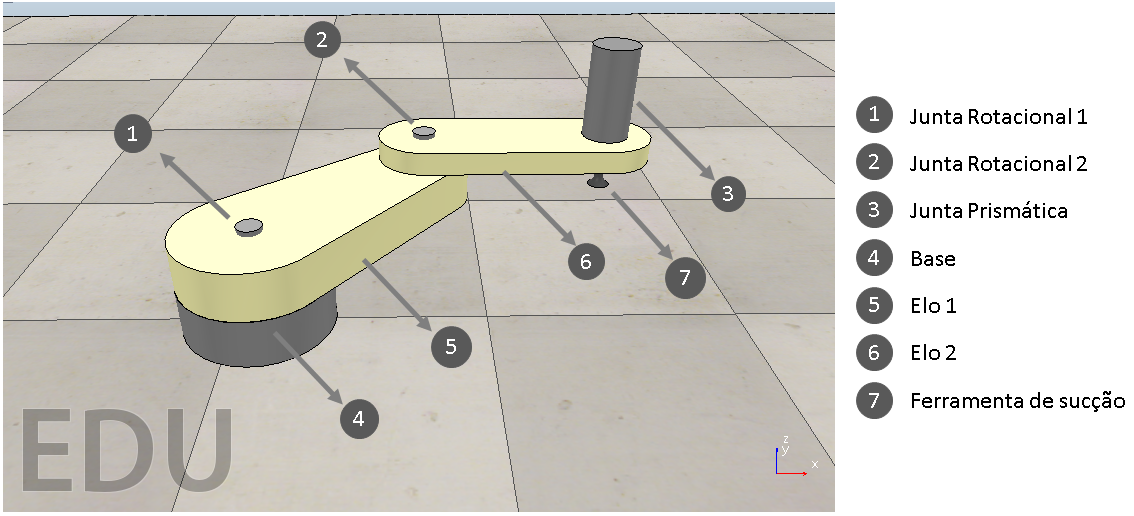
Gerando dois conjuntos de vetores para os elos, uma para cada configuração possível, pode-se realizar a plotagem dos dois simultaneamente e verificar se ambos alcançam de fato o mesmo ponto no espaço. O teste realizado é observado na figura X abaixo.

Cada um dos conjuntos de linhas vermelhas representa uma configuração possível de aceso ao ponto final dentro do workspace do robô SCARA. Com o teste de posicionamento foi possível verificar graficamente a veracidade do modelo desenvolvido, a próxima etapa é aplicar o modelo ao hardware embarcado no Arduino.

* 1. **MODELO TRIDIMENSIONAL**

A simulação foi realizada com um modelo tridimensional previamente instalado com o software V-Rep, provido pela *Coppelia Robotics.* O modelo conta com todos os elementos necessários para simular um modelo simples, mas completamente funcional de manipulador SCARA.

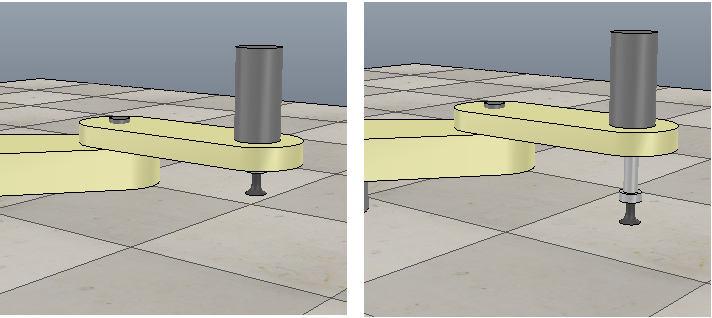
O SCARA utilizado conta com uma base e dois elos rígidos, duas juntas rotacionais, uma junta prismática e uma ferramenta de sucção, conforme observado na figura X.



Figura

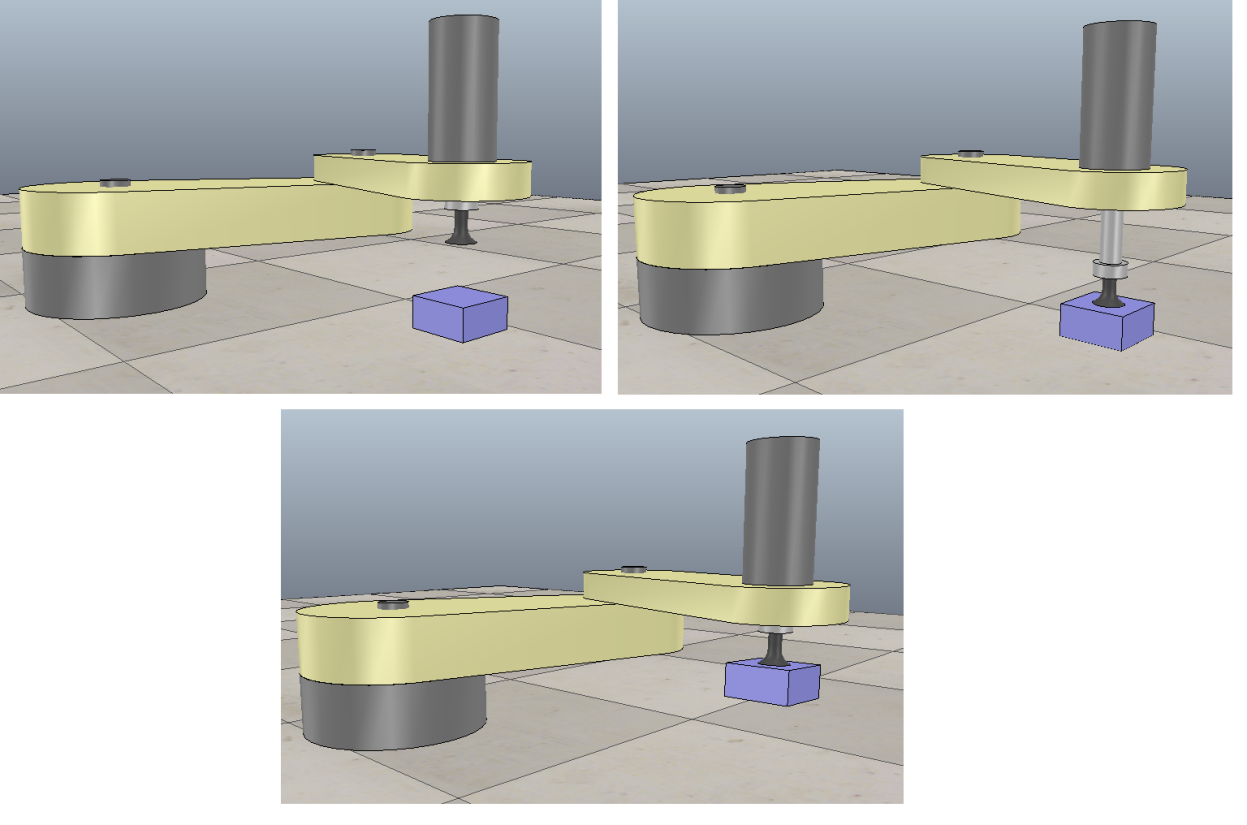
Para o robô realizar movimentos, são escritos valores dos ângulos desejados nas juntas rotacionais 1 e 2, assim é possível localizar a extremidade do robô onde desejado, e para estender ou retrair o pistão que contém a ferramenta de sucção, são escritos valores na junta prismática, para movimento linear.

A figura X abaixo apresenta a junta prismática em diferentes posições, é possível observar a possiblidade de posicionamento da extremidade do robô. O pistão é movimentado representando o grau de liberdade no eixo *z*.



Figura

Uma vez que o software executa simulações utilizando um motor de física, é possível fazer o robô interagir com outros elementos durante a simulação. Esta característica permite a simulação do funcionamento da ferramenta de sucção utilizada no modelo tridimensional. A figura X a seguir demonstra o funcionamento da ferramenta de sucção.



Figura

* 1. **MODELAGEM EM ARDUINO**